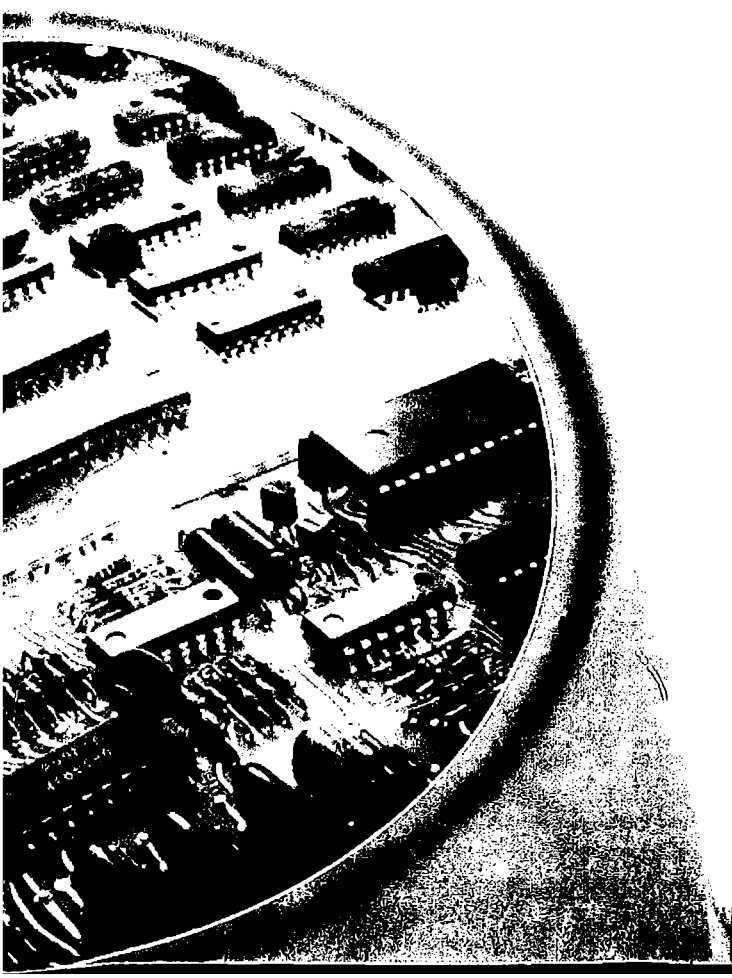
CARMEN POPESCU

Culegere de probleme

de

INFORMATICĂ



DONARI

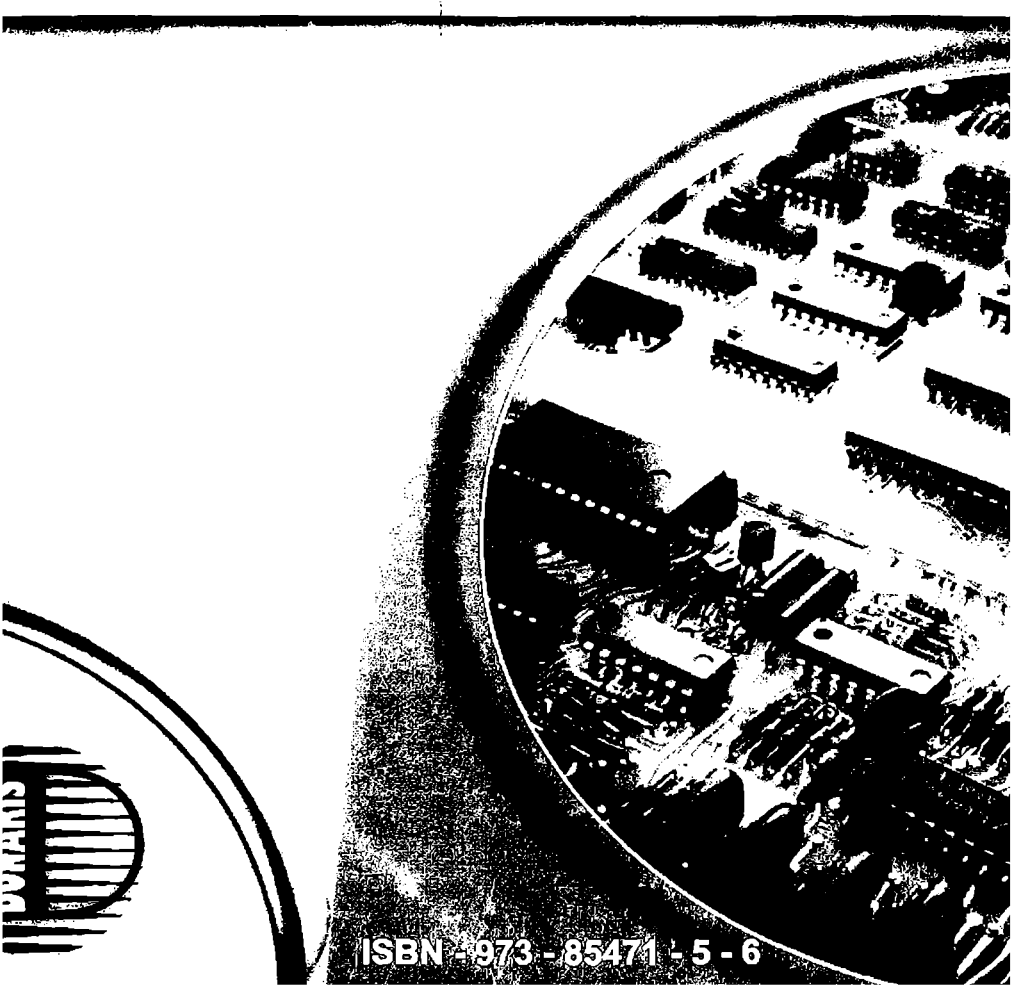
În colecția "Donaris-Info" vă mai propunem:

George Daniel Mateescu, Pavel Florin Moraru
"Informatica pentru Liceu şi Bacalaureat
- Materia de clasa a IX-a, varianta Pascal"

George Daniel Mateescu, Pavel Florin Moraru
"Informatica pentru Liceu şi Bacalaureat
- Materia de clasa a X-a, varianta Pascal"

George Daniel Mateescu, Pavel Florin Moraru, Carmen Popescu - *"Informatica - Teste şi variante de subiecte pentru Bacalaureat"*

Cristina Dragomirescu, Laura Dora Satnoianu, Mirela Popa
- "Informatica, Fişe de lucru pentru laborator - Clasa a XI -a"



ISBN - 973 - 85471 - 5 - 6

CARMEN POPESCU

CULEGERE DE PROBLEME

de

INFORMATICĂ

5 Colecția
Donaris – Info

Copyright © 2002 – Editura Donaris, Sibiu
Toate drepturile sunt rezervate autorilor şi Editurii Donaris.
Nici o parte a acestei lucrări nu poate fi copiată sau reprodusă
fără permisiunea scrisă a Editurii Donaris.
Printed in Romania

Culegere, tehnoredactare computerizată şi design: Carmen Popescu

Coperta colecţiei "Donaris Info": Ciprian Ciocan

Comenzile se pot face prin:

S.C. "INFOCART" S.R.L. – EDITURA DONARIS – SIBIU

Str. Rahovei nr.17, Sibiu, cod 2400, Jud. Sibiu

Tel: 0269 / 42.33.50

Mobile: 0744 / 69.93.03, 0745 / 31.88.47, 0745 / 534266, 0723 / 25.39.48.

E-Mail: donaris@rdslink.ro

I.S.B.N.: 973-85471-5-6

Tiparul: *Masik* Sibiu; Tel. & Fax: 0269/23.11.10

Cuvânt înainte

V-aţi putea desigur întreba, încă de la început, la ce bună o nouă culegere de informatică? Oare merită să o am în bibliotecă? Am nevoie de ea? Sau va fi doar încă o carte pe care să se depună în timp praful.

Răspunsul, vă asigur, este unul în favoarea acestei cărţi. Începând cu numărul mare de probleme, cu diversitatea acestora, cu modul de organizare, şi mergând până la design-ul, sperăm noi atractiv, totul a fost gândit cu scopul de a atrage şi ajuta elevii, şi nu numai, în dorinţa lor de a pătrunde tainele programării sau de a-şi perfecţiona tehnica rezolvării de probleme.

Culegerea cuprinde un număr de 1150 de probleme, organizate pe capitole şi paragrafe, respectând în mare structura programei şcolare valabilă pentru clasele de matematică-informatică.

Un lucru destul de important la această culegere este acela că la aproape toate problemele propuse s-au oferit şi exemple numerice, pentru a ajuta cititorul să înţeleagă mai bine cerinţele problemei, precum şi pentru a uşura verificarea corectitudinii rezolvării.

Culegerea se va dovedi un auxiliar extrem de util profesorului de informatică. Acesta va avea la dispoziţie o gamă largă de probleme pe care să le folosească în munca la clasă, în laborator, sau se pot constitui ca material de studiu individual pentru elevi.

Înainte de a porni la drum, doresc să menţionez sprijinul pe care colegii mei Georgeta Preda, Antoniu Pitic, Pavel Florin Moraru, mi l-au acordat în realizarea acestei lucrări. Fără de ei această carte ar fi fost poate mai săracă. Nu pot să nu-i amintesc şi pe elevii mei din clasele a XI-a I şi a XII-a G de la Colegiul Naţional "Gheorghe Lazăr" Sibiu, pe care i-am "chinuit" cu o parte din aceste probleme, reuşind astfel să modific, adaptez şi exemplific multe din problemele propuse.

Dorindu-vă doar mesaje "*Compile successful*", vă propun să realizăm o legătură directă autor-cititor, orice sugestie şi propunere referitoare la conţinut, formă, sau design fiind bine venite pe adresa editurii.

prof. Carmen Popescu

Cuprins

CUVÂNT ÎNAINTE 3

1. ELEMENTE DE BAZĂ ALE LIMBAJULUI 7

2. INSTRUCȚIUNI DE CICLARE 17

2.1. Divizibilitate 17

2.2. Cifrele unui număr 21

2.3. Generări de şiruri 28

2.4. Calcule de sume şi produse 31

2.5. Probleme diverse 34

3. TABLOURI 45

3.1. Tablouri unidimensionale 45

3.1.1. Parcurgeri (sume, numărări) 45

3.1.2. Maxime şi minime 64

3.1.3. Căutări de elemente 68

3.1.4. Subşiruri de elemente 70

3.1.5. Sortări şi interclasări 72

3.2. Tablouri bidimensionale (matrice) 76

3.2.1.Parcurgeri (sume numărări) 76

3.2.2. Maxime şi minime 92

3.2.3. Căutări de elemente 96

3.2.4. Sortări şi interclasări 97

4. ALTE TIPURI STRUCTURATE DE DATE 101

4.1. Tipul string 101

4.2. Tipul mulțime (set of) 111

4.3. Tipul înregistrare (record) 115

4.4. Tipul fişier 120

5. SUBPROGRAME 139

6. RECURSIVITATEA 157

6.1. Funcţii recursive 157

6.2. Proceduri recursive 165

6.3. Recursivitatea indirectă 173

7. METODE DE PROGRAMARE 175

7.1. Backtracking obişnuit 175

7.2. Backtracking generalizat 204

7.3. Metoda "Divide et Impera" 217

7.4. Metoda Greedy 219

7.5. Metoda programării dinamice 220

8. STRUCTURI DINAMICE DE DATE 231

8.1. Liste 231

8.1.1. Liste simplu înlănţuite 231

8.1.2. Liste circulare 243

8.1.3. Liste liniare dublu înlănţuite 244

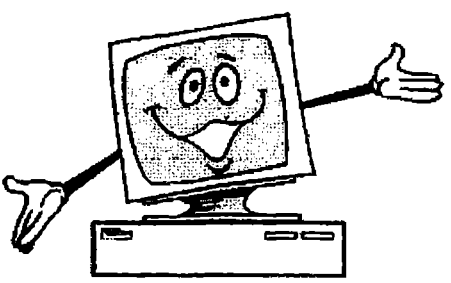
8.1.4. Structuri diverse de tip listă 246

8.2. Arbori 249

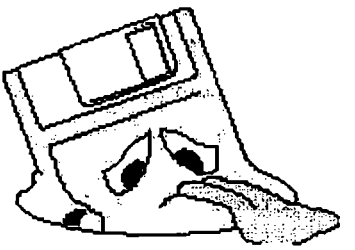
8.2.1. Arbori binari 249

8.2.2. Arbori oarecare 256

BIBLIOGRAFIE 257



1. Elemente de bază ale limbajului



1. Să se scrie un program care să calculeze numărul de picioare dintr-o curte, în care se află `g` găini, `p` pisici şi un om.

   Exemplu. Pentru `g=5`, `p=3` se va afişa `24` picioare.

2. Să se determine ultima cifră a sumei `x+y`, unde `x` şi `y` sunt numere naturale date de la tastatură.

   Exemplu. Pentru `x=123` şi `y=349` se va afişa `2`.

3. Să se calculeze aria unui triunghi cunoscând laturile `a`, `b`, `c` ale triunghiului.

   Exemplu. Pentru `a=2`, `b=3`, `c=4` se va afişa `s=2.90473`, iar pentru `a=3`, `b=4`, `c=5` se va afişa `s=6`.

4. Să se calculeze aria unui triunghi cunoscând o latură a triunghiului şi înălţimea corespunzătoare.

   Exemplu. Pentru `l=4` şi `h=3` se va afişa `s=6`.

5. Un melc se deplasează cu viteza `v` km/săptămână. De cât timp (în ore) are nevoie melcul pentru a străbate distanţa `d` dată în metrii.

   Exemplu. Pentru `v=3` km/saptamana şi `d=5` m se va afisa `t=0.28` ore.

6. Fie `a` şi `b` două variabile reale. Să se interschimbe valorile celor două variabile:
   a) folosind o variabilă auxiliară;
   b) fără a folosi variabile auxiliare.

7. Perimetrul unui pătrat este egal cu latura altui pătrat. Ştim că suma perimetrelor este `x`. Să se calculeze ariile celor două pătrate.

   Exemplu. Pentru `x=20` se va afişa: `aria1=1`, `aria2=16`.

8. Fiind date trei numere naturale nenule `a`, `b` şi `k`, să se verifice dacă fracţia `a/b` se simplifică prin `k`. În caz afirmativ se va afişa şi fracţia simplificată.

   Exemplu. Fracţia `36/48` se simplifică cu `4` şi se afişează fracţia `9/12`, iar fracţia `12/89` nu se simplifică cu `4`.

9. Se citeşte de la tastatură un număr reprezentând o zi calendaristică dintr-o lună oarecare. Să se afişeze decada căreia îi aparţine ziua.

   Exemplu. Data de `13` este în a doua decadă iar `25` este în a treia (respectiv ultima) decadă.

10. Să se verifice dacă un număr natural este sau nu pătrat perfect.

    Exemplu. `64` (`64=8*8`) este pătrat perfect `35` nu este pătrat perfect

11. Să se verifice dacă un număr natural este sau nu cub perfect.

    Exemplu. `125` este cub perfect (`125=5*5*5`).

12. Scrieţi un program care calculează şi afişează volumul unui trunchi de con $V = \frac{\pi \cdot h}{3}\left(R_1^2 + R_2^2 + R_1 \cdot R_2\right)$. Condiţii: `R1, R2, h > 0`.

    Exemplu. Dacă `R1=2, R2=7, h=4` atunci `V=280.648944`.

13. Să se scrie un program pentru calculul ariei totale a unui cilindru circular drept $A = 2 \cdot \pi \cdot R\ (R + G)$. Condiţii: `R, G > 0`.

    Exemplu. `R=2, G=9` atunci `A=138.23008`.

14. Aria unui trapez oarecare $A = \frac{(b1 + b2)\ h}{2}$. Condiţii: `b1, b2, h > 0`.

    Exemplu. `b1=2, b2=9, h=4` => `A=22`.

15. Se citeşte de la tastatură o cifră între `1` şi `7` reprezentând o zi din săptămână. Să se afişeze denumirea zilei. Zilele săptămânii sunt numerotate astfel: `1→luni,2→marţi,...,7→duminică`.

16. Se citeşte de la tastatură o literă. Se cere să se verifice dacă ea este vocală sau consoană (considerăm alfabetul englez în care vocalele sunt `'a', 'e', 'i', 'o', 'u'`).

17. Se citesc de la tastatură două numere întregi, `a`, `b` şi un operator `op` dintre `+, -, *, /`. Se cere să se afişeze valoarea asociată expresiei `a op b`.

    Exemplu. `a=5, b=12, op='*'` se va afişa `60`.

18. O navă parcurge distanţa `d` (dată în kilometrii) în `a` săptămâni şi `b` zile. Care este viteza în `m/s`?

    Exemplu. Pentru `a=10` săptămâni, `b=5` zile şi `d=1000` kilometrii se va afişa: `viteza=0,15432 m/s`.

19. Să se scrie un program care să calculeze suma a două intervale de timp exprimate în ore, minute, secunde şi zecimi de secunde.

    Exemplu. Pentru `t1= 5 ore 45 min 36 sec 20 zecimi` şi
    `t2= 1 oră 30 min 29 sec 90 zecimi`

    Se va afişa: `7 ore 16 min 6 sec 10 zecimi`.

20. Se dau două numere naturale `a` şi `b` cu cel mult trei cifre. Să se afişeze numărul care are suma cifrelor mai mare.

    Exemplu: daca `a=321` iar `b=263` se va afişa `263`.

21. Se dau două numere **a** şi **b**. Să se calculeze şi afişeze maximul celor două valori.

Exemplu. pentru **a=30** şi **b=24** se va afişa **30**

pentru **a=2** şi **b=32** se va afişa **32**

22. Se dau trei numere **a**, **b** şi **c**. Să se determine maximul şi minimul acestor valori.

Exemplu: Pentru **a=34, b=12, c=43** se va afişa **minim=12** şi **maxim=43.**

23. Să se afişeze minimul a patru numere date **a**, **b**, **c**, **d**.

Exemplu: Pentru **a=23, b=224, c=4, d=45** minimul este **4**.

24. Se citesc numerele întregi **a**, **b**, **c**, **d**. Să se afişeze maximul valorilor negative şi minimul valorilor pozitive dintre **a**, **b**, **c**, **d**.

Exemplu. Dacă **a=-4, b=-2, b=12, c=5** atunci minimul valorilor pozitive este **5** iar maximul valorilor negative este **-2**.

25. Scrieţi un program care va citi o singură literă de la tastatură şi va afişa cifra corespunzătoare de pe tastele unui telefon. Cifrele şi literele de la telefon sunt asociate după cum urmează:

2=A, B, C	**3=D, E, F**	**4=G, H, I**
5=J, K, L	**6=M, N, O**	**7=P, R, S**
8=T, U, V	**9=W, X, Y**	

Nu există nici o cifră care să corespundă literei **Q** sau **Z**. Pentru aceste litere programul va afişa un mesaj indicând că ele nu există pe tastatura telefonului. De asemenea cifrele **0** şi **1** nu au asociate nici o literă.

26. Să se verifice dacă un an calendaristic dat este bisect sau nu. Un an este bisect dacă
    - este divizibil cu **4** ŞI nu este divizibil cu **100** SAU
    - este divizibil cu **400**.

Exemplu. Anii **2000**, **1984** au fost ani bisecţi iar anii **1900**, **1975** nu au fost bisecţi.

27. Se dau numerele **z,l,a∈N**. Să se determine dacă tripletul **(z,l,a)** reprezintă o dată calendaristică corectă a secolului **XX**.

Exemplu. Data de **31.04.1953** nu este o dată corectă a secolului **XX** (luna a patra are doar **30** de zile). Data de **29.02.1990** nu este o dată corectă a secolului **XX** (anul **1990** nu este an bisect).

28. Să se verifice dacă trei numere naturale date **a**, **b** şi **c** sunt pitagorice (adică dacă verifică una dintre condiţiile $a^2=b^2+c^2$, $b^2=a^2+c^2$, $c^2=a^2+b^2$).

Exemplu: numerele **3 4 5** sunt pitagorice.

29. Se cunosc numele şi media generală la sfârşitul anului şcolar a trei elevi. Să se afişeze numele acestora în ordine descrescătoare a mediei generale.

Exemplu: Daca numele elevilor si mediile corespunzătoare sunt:

```
Ionescu 6,80
Antonescu 9.40
Popescu 8.25
```

se va afişa `Antonescu, Popescu, Ionescu`

30. Se citeşte o sumă de bani S. Să se tipărească numărul minim de monede de tipurile `T1, T2, T3` pentru a plăti suma respectivă. *Observaţie:* Se consideră că problema are întotdeauna soluţie.

Exemplu. `t1=20; t2=100; t3=1; s=120` se va afişa `1` monedă de `100; 1` monedă de `20`.

31. Se dau trei numere `a, b, c`. Să se afişeze cele trei numere în ordine crescătoare.

a) nu se vor modifica valorile iniţiale ale variabilelor;

b) valorile iniţiale se vor modifica astfel încât după rularea programului `a<b<c`.

Exemplu: daca `a=85 b=34 c=67` se va afişa `34, 67, 85`

32. Scrieţi un program de convertire a unităţilor de măsură din sistemul englez în sistemul metric sau invers. Unităţile luate în calcul vor fi:

```
1. Inch (in)   3. Pound (lb)     5.Gallon (gal)
2. Foot (ft)   4. Ounce (oz)
```

respectiv

```
1. Centimetru (cm)  3. Kilogram (kg)  5. Litru (l)
2. Metru (m)        4. Gram (g).
```

Vor fi utilizaţi următorii factori de conversie

```
1 in = 2,54 cm    1 lb = 0,45359 kg
1 gal=3,78541 l   1 ft = 0,3048 m   1 oz = 28,3495 g
```

Exemplu. `13 in = 33.02 cm = 0.3302 m.`

33. Se dau trei variabile întregi `a, b, c` a căror valori se citesc de la tastatură. Se cere să se permute circular spre stânga valorile acestor variabile şi apoi să se afişeze.

Exemplu: Dacă numerele citite sunt `a=1; b=2; c=3` dupa rularea programului acestea vor avea următoarele valori: `a=2; b=3; c=1`.

34. Se citeşte de la tastatură un număr de patru cifre. Se cere să se verifice dacă numărul este palindrom, adică dacă citindu-l de la sfârşit spre început se obţine acelaşi număr.

Exemplu: **1331** este palindrom.

35. Data pentru orice zi de Paşti poate fi calculată după cum urmează

$$A = \left\{\frac{An}{19}\right\} \qquad D = \left\{\frac{19\ \ A + 24}{30}\right\}$$

$$B = \left\{\frac{An}{4}\right\} \qquad E = \left\{\frac{2\ \ B + 4\ \ C + 6\ \ D + 5}{7}\right\}$$

$$C = \left\{\frac{An}{7}\right\}$$

unde **{a/b}** este restul împărţirii lui **a** la **b**. Atunci data pentru Paşti este **(22+E+D) Martie**. Observăm că această formulă poate da o zi din aprilie dacă **22+E+D>31**. *Observaţie.* Nu se au în vedere diferenţele apărute din decalajul dintre Paştele Catolic şi cel Ortodox. (luaţi problema ca un simplu exerciţiu de programare ☺)

Exemplu. **An=1985** => Paştele sunt în **7 Aprilie 1985**.

36. Să se rezolve ecuaţia $ax^2+bx+c=0$ unde **a, b, c** sunt numere reale citite de la tastatură. Se vor trata toate cazurile posibile.

Exemplu: dacă **a=1, b=1, c=1** se va afişa "sistemul nu are soluţii reale"

dacă **a=1, b=-2, c=1** se va afişa **x1=x2=1**

dacă **a=1, b=-3, c=2** se va afişa **x1=1, x2=2**

(Atenţie: soluţiile pot fi reale chiar dacă coeficienţii sunt numere întregi)

37. Să se calculeze diferenţa a două unghiuri date în grade, minute şi secunde.

Exemplu. **30° 50' 45'' - 10° 20' 53'' = 20° 29' 52''**

38. Să se determine ultimele două cifre ale produsului **a*b**, pentru **a** şi **b** citite de la tastatură.

Exemplu. Dacă **a=15** iar **b=34** se vor afişa cifrele **1** şi **0**.

39. Să se verifice dacă patru numere **a, b, c, d** date de la tastatură pot forma o mulţime.

Indicaţie. O mulţime este formată din elemente distincte două câte două!

40. Se citesc trei numere întregi **a, b, c**. Să se verifice dacă aceste numere (puse în orice ordine) sunt în progresie aritmetică şi afişaţi raţia progresiei în caz afirmativ.

Exemplu. **(10, 3, 17)** este o progresie cu raţia **7**; **(1, 2, 4)** nu este progresie aritmetică.

41. Se dau trei numere reale **a**, **b**, **c**. Pot reprezenta ele lungimile unui triunghi? În caz afirmativ să se precizeze de ce tip este triunghiul: echilateral, isoscel, dreptunghic, sau oarecare.

Exemplu. **(1, 2, 3)** – nu este triunghi; **(3, 4, 5)** – triunghi dreptunghic; **(2, 2, 2)** – triunghi echilateral; **(3, 6, 6)** – triunghi isoscel.

42. Se dau trei puncte în plan $M_i=(x_i,y_i)$, **i=1,2,3**. Să se verifice dacă ele sunt coliniare și să se afișeze un mesaj corespunzător.

Exemplu. **(1,2), (7,12), (4,7)** – sunt coliniare; **(1,2), (1,4), (3,4)** – nu sunt coliniare.

43. Se citesc de la tastatură trei numere reale **a**, **b**, **c**. Să se verifice dacă acestea pot fi unghiuri ale unui triunghi exprimate în grade, iar în caz afirmativ să se determine natura triunghiului (optuz-unghic, dreptunghic sau ascuțit-unghic).

Exemplu. **(0,80,100)** – nu e triunghi; **(25,15,140)** – triunghi optuz-unghic; **(90,30,60)** – triunghi dreptunghic; **(55,40,85)** – triunghi ascutit-unghic.

44. Un punct în plan este dat prin coordonatele sale **(x,y)**. Să se scrie un program care determină dacă punctul este în origine, într-un cadran (**1**,**2**,**3** sau **4**), sau pe una din semiaxe (**Ox**, **Ox'**, **Oy**, **Oy'**).

Exemplu. **(1,1)** – cadranul **1**; **(0,3)** – axa **Oy**; **(-2,4)** – cadranul **2**.

45. Să se rezolve ecuația $\sqrt{ax+b}=c$ cu **a**, **b**, **c** numere reale citite de la tastatură. (discuție)

Exemplu. **a=2, b=5, c=-1** => ecuație imposibilă; **a=0, b=2, c=4** => o infinitate de soluții; **a=5, b=6, c=9** => **x=15**.

46. Să se rezolve următorul sistem de ecuații $\begin{cases} ax+by=c \\ dx+ey=f \end{cases}$, cu **a**, **b**, **c**, **d**, **e**, **f** numere reale nenule. Discuție.

Exemplu. **a=1, b=4, c=3, d=-10, e=2, f=15** => **x=-1.186046, y=1.046511**; **a=2, b=5, c=-37, d=14, e=35, f=-259** => o infinitate de soluții; **a=1, b=5, c=1, d=3, e=15, f=2** => sistemul nu are soluții.

47. Se consideră punctele $A(x_1,y_1)$, $B(x_2,y_2)$, $C(x_3,y_3)$, $D(x_4,y_4)$. Citindu-se coordonatele acestor puncte să se stabilească dacă cele patru puncte sunt vârfurile unui dreptunghi sau nu. Punctele se vor da parcurgând laturile patrulaterului (convex) în ordine trigonometrică.

Exemplu. **A(10,4), B(5,9), C(13,7), D(8,2)** formează un dreptunghi.

48. Să se calculeze suma $S = 1 - 2 + 3 - 4 + \ldots \pm n$, pentru **n** dat de la tastatură. Nu se vor folosi nici un fel de instrucțiuni repetitive.

Exemplu. Dacă **n=10** se obține suma **-5**.

49. Se citesc patru perechi de numere reale (x_1,y_1), (x_2,y_2), (x_3,y_3) şi (x_p,y_p). Să se decidă dacă punctul de coordonate (x_p,y_p) este în interiorul triunghiului format prin unirea primelor 3 puncte.

Exemplu. Dacă triunghiul are vârfurile (7,8), (12,3), (2,1) atunci punctele (3,6), (9,1), (10,7) sunt exterioare triunghiului iar punctele (4,2) şi (10,5) sunt interioare triunghiului.

50. Să se scrie programul pentru transcrierea în cuvinte a unei valori întregi de patru cifre, introduse de la tastatură.

Exemplu.

1890: o mie opt sute nouăzeci

1821: o mie opt sute două zeci şi unu

1818: o mie opt sute optsprezece

1800: o mie opt sute

1000: o mie

51. Să se calculeze ultima cifră a numărului 2^x, pentru x număr natural, fără a calcula efectiv puterea.

Exemplu. Ultima cifra a numărului 2^{259} este **8**.

52. Să se determine ultima cifră a numărului a^b, pentru **a** şi **b** numere naturale mai mici sau egale cu **32000**, citite de la tastatură.

Exemplu. Numărul 124^{59} se termină în cifra **4**.

53. Să se calculeze ultima cifră a sumei $2^a + 3^a + \ldots + 9^a$, fără a calcula efectiv suma.

Exemplu. Ultima cifră a sumei pentru **a=2379** este **4**.

54. Se citeşte o succesiune de condiţii şi instrucţiuni scrise liniar. Ştiind că fiecare condiţie are cel mult două cazuri şi în fiecare caz apare o singură instrucţiune, să e construiască structurile posibile ale algoritmului.

Exemplu. **C1 C2 C3 I1 I2 C4 I3 =>**

```
daca C1 atunci
  daca C2 atunci
    daca C3 atunci
      I1
    altfel
      I2
   sf. daca
  altfel
    daca C4 atunci
      I3
    sf. daca
  sf. daca
```

55. Să se calculeze valoarea funcției matematice **f(x)**, pentru o valoare a lui **x** introdusă de la tastatură:

$$f: R \to R, \quad f(x) = \begin{cases} x^2 + 1 & \text{pentru } x \le -3 \\ x - 2 & \text{pentru } x \in (-3, 3). \\ x^2 - 4x + 5, & \text{pentru } x \ge 3 \end{cases}$$

Exemplu. **f(-4.54)=21.6116; f(1.25)=-0.75; f(5.5)=13.25.**

56. Se citesc patru numere întregi **a, b, c, d.** Dacă cel puțin trei sunt strict pozitive să se calculeze:

$$x = \begin{cases} a + b + c + d & \text{daca suma celor strict pozitive} \\ & \text{(din cele 4 numere) este para} \\ a \cdot b \cdot c \cdot d & \text{in caz contrar} \end{cases}$$

Exemplu. **a=1, b=6, c=3, d=5 => x=90; a=-5, b=9, c=7, d=-4 => x=7.**

57. Se dau **a, b, c, d, e** valori logice și **x, y, z** numere reale. Să se calculeze:

$$v = \begin{cases} \max\{x, y, z\} & \text{daca predomina valoarea true} \\ \min\{x, y, z\} & \text{daca predomina valoarea false} \end{cases}$$

Exemplu. **a=e=c=true, b=d=false, x=3, y=9, z=1 => v=9**

a=e=c=false, b=d=true, x=3, y=9, z=1 => v=1.

58. Se citesc **4** valori reale **a, b, c, d.** Să se evalueze expresia

$$F = \begin{cases} a + b & \text{daca } c + d > 0 \\ a - b & \text{daca } c + d = 0. \\ a * b & \text{daca } c + d < 0 \end{cases}$$

Exemplu. **a=-1.4, b=6.1, c=5, d=-20 => F=-8.54;**

a=-1.4, b=6.1, c=-5, d= 20 => F= 4.7;

a=-1.4, b=6.1, c=5, d=-5 => F=-7.5.

59. Să se calculeze valoarea funcției **F** definită pe mulțimea numerelor reale, pentru **x** citit:

$$F = \begin{cases} x^2 - 3 & \text{daca } x < 5 \\ x + 1 & \text{daca } 5 \le x \le 25. \\ x^2 - 5x + 6 & \text{daca } x > 25 \end{cases}$$

Exemplu. **x=-6.5 => f=39.25; x=5.75 => f=6.75; x=27.5 => f=624.75.**

60. Pentru fiecare dintre funcțiile matematice de mai jos, să se scrie câte un program care afișează valoarea funcției pentru o valoare a argumentului x citită de la tastatură:

a) $f(x) = \begin{cases} x + 1, & \text{daca } 0 < x < 1 \\ x^2 - 5 \quad x + 6, & \text{altfel} \end{cases}$

Exemplu. x=0.5 => f=1.5; x=4 => f=2; x=0 => f=6.

b) $f(x) = \begin{cases} \sqrt{x^2 + x + 2} & \text{pentru } x < 1 \\ 3x - 1, & \text{pentru } 1 \le x \le 2 \\ \dfrac{5x - 2}{x + 2}, & \text{pentru } x > 2 \end{cases}$

Exemplu. x=-2.5 => f=2.397915; x=1.5 => f=3.5;

x=5 => f=3.285714.

61. La o societate sunt n (n<32) acționari. Cel mai important acționar deține din capitalul societății un procent concretizat prin c milioane lei (c>1). Următorul ca importanță deține jumătate din procentul deținut de primul, următorul jumătate din procentul deținut de al doilea, etc. Realizați un program care tipărește suma deținută de ultimul acționar.

Exemplu. Dacă sunt 5 acționari iar suma deținută de primul acționar este de 130 de milioane atunci ultimul acționar deține 8.125 milioane (adică 8 milioane și ...).

62. Să se scrie un program care criptează un text citit de la tastatură, transformând fiecare caracter astfel: se inversează grupele de câte doi biți ai reprezentării în baza 2 a codului ASCII, respectiv prima grupă de doi biți cu a doua grupă și a treia cu a patra. ($b_7b_6b_5b_4b_3b_2b_1b_0 \rightarrow b_5b_4b_7b_6b_1b_0b_3b_2$).

Exemplu. ORD('X')=88

88_{10}=01 01 10 $00_2 \rightarrow$ 01 01 00 $10_2 = 82_{10}$

CHR(82)='R'

Unde am notat ORD – codul ASCII al caracterului

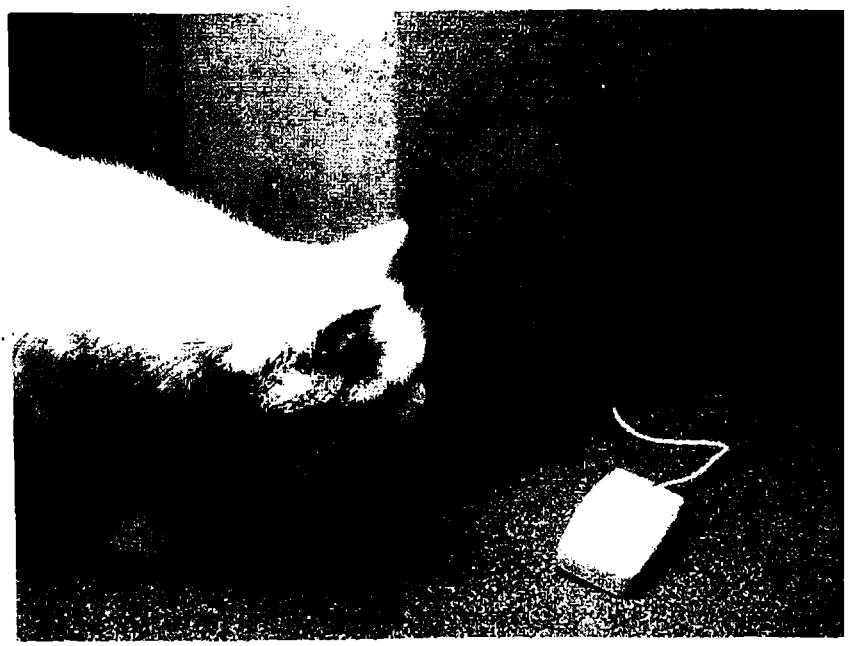
CHR – caracterul având codul ASCII precizat

63. Ținând cont că anul (considerat cu 2 cifre) se poate reprezenta pe 7 biți, luna (număr între 1 și 12) se poate reprezenta pe 4 biți și ziua (număr între 1 și 31) se poate reprezenta pe 5 biți. Se cere:

a) Să se scrie un program care împachetează o dată într-un întreg pe 16 biți astfel: primii 7 biți (cei mai reprezentativi) reprezintă anul, următorii 4 luna, iar ultimii 5 (cei mai puțin semnificativi) ziua.

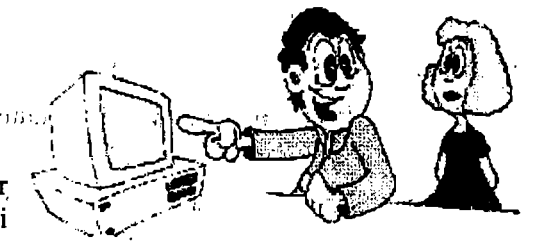
b) Să se scrie un program care despachetează data codificată ca m scriind pe ecran data sub forma obișnuită.

Exemplu. Data de `6 aprilie 2002` se codifică prin valoarea (`0000010 0100 0110`)



2. Instrucțiuni de ciclare

2.1. Divizibilitate



1. Să se afișeze suma numerelor naturale divizibile cu 3 mai mici sau egale decât o valoare dată n.

   Exemplu. Pentru n=20 se obține s=63.

2. Să se găsească toate perechile A, B de cifre astfel încât numărul $\overline{xAyB}$ să fie divizibil cu 9, unde x și y sunt cifre citite de la tastatură (x≠0).

   Exemplu. Pentru x=4 și y=1 se vor afișa perechile (0,4), (1,3), (2,2), (3,1), (4,0), (4,9), (5,8), (6,7), (7,6), (8,5), (9,4).

3. Să se determine toate numerele de forma $\overline{abcd}$ divizibile cu un n dat. Cifrele vor fi distincte două câte două.

   Exemplu. Pentru n=973 se vor afișa numerele 1946, 3892, 4865, 9730.

4. Determinați numerele de 4 cifre, divizibile cu 15 și pentru care suma primelor două cifre este egală cu 12. Câte astfel de numere există? Dar dacă se impune ca cifrele să fie distincte două câte două?

   Exemplu. Există 49 de numere, un exemplu fiind 8460. În al doilea caz sunt doar 24 de numere.

5. Să se afle toate numerele naturale mai mici decât 2000, care împărțite la 24, 30, 18 dau restul 7.

   Exemplu. 1447:24=60 rest 7; 1447:30=48 rest 7, 1447:18=80 rest 7.

6. Să se afle numerele de două cifre, care împărțite la 15 dau un rest egal cu pătratul câtului.

   Exemplu. Un astfel de număr este 34 (34:15=2 rest 4 și $2^2 = 4$).

7. Determinați toate numerele care au proprietatea că împărțind pe 80, 134 sau 152 la unul dintre ele, se obține același rest, diferit de zero.

   Exemplu. 80:6=13 rest 2; 134:6=22 rest 2; 152:6=25 rest 2.

8. Să se determine suma tuturor resturilor împărțirilor numerelor de patru cifre la 999.

   Răspuns. s=30106.

9. Să se scrie un program care generează toate numerele prime mai mici sau egale decât un număr natural citit de la tastatură.

   Exemplu. Pentru `n=10` se vor afişa numerele `2, 3, 5, 7`.

10. Să se scrie un program care generează primele `n` numere prime.

    Exemplu. Pentru `n=10` se vor afişa numerele `2, 3, 5, 7, 11, 13, 17, 19, 23, 29`.

11. Să se afişeze toate numerele prime situate în intervalul `[p,q]`, precum şi numărul acestora, unde `p` şi `q` sunt două numere naturale date.

    Exemplu. Pentru `p=10` şi `q=25` se vor afişa numerele `11, 13, 17, 19, 23`.

12. Să se afişeze toţi divizorii comuni a două numere.

    Exemplu. Divizorii comuni ai numerelor `60` şi `350` sunt `1, 2, 5, 10`.

13. Scrieţi un program care să determine cel mai mare divizor comun a două numere întregi, prin algoritmul clasic al lui Euclid (cu împărţiri repetate). Dacă numerele citite nu sunt ≥1 se cere reintroducerea lor de la tastatură.

    Exemplu. `cmmdc(882,2100)=42`.

14. Să se scrie un program care determină cel mai mare divizor comun a două numere întregi nenule `a` şi `b` folosind următoarea formulă:

$$\text{cmmdc}(a,b) = \begin{cases} \text{cmmdc}(a-b,\ b) & \text{daca } a > b \\ \text{cmmdc}(a,\ b-a) & \text{daca } b > a \\ a & \text{daca } a = b \end{cases}$$

15. Să se scrie un program care determină descompunerea în factori primi ai unui număr natural nenul dat. Afişarea se va face în forma:

```
3268 |  2
1634 |  2
 817 |  19
  43 |  43
   1 |
```

16. Fiind dat un număr natural `x`, să se afişeze factorul prim care apare la puterea cea mai mare în descompunerea lui `x` în factori primi.

    Exemplu. Pentru `x=1620`, descompunerea este $2^2 \cdot 3^4 \cdot 5$, iar factorul cerut este `3`.

17. Să se determine dacă două numere sunt prime între ele sau nu. Două numere naturale sunt prime între ele dacă cel mai mare divizor comun al lor este `1`.

    Exemplu. `15` şi `38` sunt numere prime între ele.

18. Să se scrie un program care determină cel mai mic număr care are exact **k** divizori.

Exemplu. Pentru **k=4** se obține numărul **6**.

19. Două numere prime impare consecutive se numesc numere prime gemene. Determinați toate perechile de numere prime gemene **≤100**.

Exemplu. O pereche de numere prime gemene este **(71,73)**.

20. Să se afișeze primele **n** perechi de numere prime gemene.

Exemplu. Pentru **n=5** perechile obținute sunt **(3,5)**, **(5,7)**, **(11,13)**, **(17,19)**, **(29,31)**.

21. Un număr natural se numește perfect dacă el este egal cu suma divizorilor săi mai mici decât el. Să se verifice dacă un număr **n** este perfect sau nu.

Exemplu. Pentru **n=28** este număr perfect deoarece **28=1+2+4+7+14**.

22. Să se determine toate numerele perfecte mai mici decât **10000**.

23. Se citește un șir de numere întregi până când se introduce de două ori consecutiv aceeași valoare. Să se afișeze câte pătrate perfecte sunt în șir.

Exemplu: dacă șirul este **13 9 56 400 8 17 25 25**, acesta conține patru pătrate perfecte (numerele **9**, **400**, **25** și **25**).

24. Determinați cel mai mic număr **≤n** care are numărul maxim de divizori proprii (divizorul propriu este diferit de **1** și de numărul însuși).

Exemplu. Pentru **n=20** se obține numărul **12** (numerele 18 și 20 au același număr de divizori proprii).

25. Pentru un număr natural **n** citit de la tastatură, **1≤n≤1000** se va afișa mulțimea numerelor naturale **<n** care sunt prime cu **n**. Se va afișa de asemenea și numărul acestor numere (notat **φ(n)**).

Exemplu. Pentru **n=20** se afișează numerele **3**, **7**, **9**, **11**, **13**, **17**, **19**, **φ(20) = 7**.

26. Pentru un număr **n** citit de la tastatură **1≤n≤1000**, se va afișa mulțimea divizorilor săi naturali (inclusiv **1** și **n**). De asemenea se vor afișa numerele:

τ(n) = numărul divizorilor lui **n**

σ(n) = suma divizorilor lui **n**.

Exemplu. Divizorii lui **n=20** sunt **1**, **2**, **4**, **5**, **10**, **20**, suma lor este **42**, iar numărul lor este **6**.

27. Să se scrie un program care să calculeze câte perechi de numere naturale care nu depășesc un număr natural dat au cel mai mare divizor comun un număr dat **d**.

Exemplu. Pentru **n=20** există **6** de perechi **(5,5)**, **(5,10)**, **(5,15)**, **(5,20)**, **(10,15)**, **(15,20)** care au cel mai mare divizor comun egal cu **5**.

28. Se citesc de la tastatură **n** fracții în forma (numărător, numitor). Se cere să se calculeze suma acestor fracții (în formă ireductibilă).

Exemplu. $\frac{7}{6} + \frac{1}{3} + \frac{1}{4} + \frac{2}{5} = \frac{43}{20}$.

29. Se citesc **n** numere naturale de la tastatură. Să se determine cu câte zerouri se sfârșește produsul acestora, fără a efectua produsul.

Exemplu. Produsul **23·48·15·25·34** se termină cu **3** zerouri.

30. Se citesc pe rând **n** numere naturale și un număr prim **p**. Se cere să se găsească **k** maxim, astfel încât p^k să dividă produsul celor n numere naturale. Se va evita efectuarea produsului celor **n** numere.

Exemplu. Pentru **n=5**, **p=2**, și numerele **10**, **2**, **19**, **32**, **174** valoarea lui **k** maxim este **8**.

31. Pentru un număr natural **n** citit de la tastatură, să se afișeze tripletele **(x,y,z)** de numere naturale cu $1 \le x < y < z \le n$, care sunt pitagorice, adică $x^2 + y^2 = z^2$ și **cmmdc(x,y)=1**.

Exemplu. Pentru **n=20** se afișează tripletele **(3,4,5)**, **(5,12,13)**, **(8,15,17)**.

32. Două numere întregi **x** și **y** sunt prietene dacă suma divizorilor numărului **x** este egală cu suma divizorilor numărului **y**. Să se găsească toate numerele prietene din intervalul **[a,b]**.

Exemplu. Pentru **a=10** și **b=25** se vor afișa perechile **(10,17)**, **(14,15)**, **(14,23)**, **(15,23)**, **(16,25)**.

33. (Conjectura lui Goldbach.) Orice număr natural par mai mare decât **4** se poate scrie cu sumă de numere prime impare. Verificați această conjectură pentru numere mai mici sau egale cu **1000**.

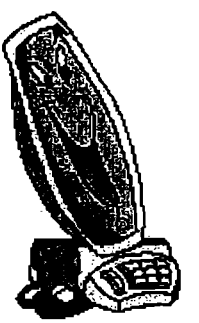
Exemplu. Pentru **n=292** putem scrie **n=283+7+2**.

34. Să se calculeze exponentul la care apare numărul prim **p** în descompunerea numărului **1·2·...·n** fără a efectua înmulțirea.

Exemplu. Pentru **n=20** și **p=3** se obține exponentul **8**.

35. Calculați **(A^B) mod C**, unde **0<=A,B,C<=MaxLongInt**.

Exemplu. Pentru **A=2**, **B=5**, **C=3** se va afișa **2**.



2.2. *Cifrele unui număr*

36. Se citeşte un număr întreg. Să se convertească într-un şir de caractere astfel cifra `0` este înlocuită de caracterul '`a`', cifra `1` de caracterul '`b`', s.a.m.d.

    Exemplu. Pentru `n=4529` se obţine şirul '`efcj`'.

37. Se dă un număr natural cu cel mult `9` cifre. Să se afle numărul de cifre pare.

    Exemplu. Numărul `236461` are `4` cifre pare.

38. Care sunt numerele prime de trei cifre care au produsul cifrelor egal cu o valoare `p` dată.

    Exemplu. Pentru `p=9` numerele `191, 313, 331, 911` îndeplinesc condiţiile problemei.

39. Să se găsească toate numerele formate din cinci cifre, care îndeplinesc simultan următoarele condiţii:
    a doua cifră este egală cu de `4` ori prima cifră;
    ultima cifră este egală cu a doua cifră
    a treia cifră reprezintă produsul dintre a patra şi a cincea.

    Exemplu. Un astfel de număr este `28008`.

40. Se cere să se afişeze toate numerele de `3` cifre având cifrele în ordine crescătoare şi suma lor este egală cu `18`.

    Exemplu. Un astfel de număr este `369`.

41. Să se găsească numerele de două cifre care au următoarea proprietate: răsturnatul pătratului numărului este egal cu pătratul numărului răsturnat.

    Exemplu. Un număr care îndeplineşte condiţiile problemei este `13` (`13`^{2}`=169`, `31`^{2}`=691`).

42. Să se afişeze toate perechile de numere palindromice din intervalul `[a,b]`. O pereche de numere se numeşte palindromică dacă al doilea este răsturnatul primului.

    Exemplu. Pentru `a=10` şi `b=40` se obţin perechile `(11,11)`, `(12,21)`, `(13,31)`, `(22,22)`, `(23,32)`, `(33,33)`.

43. Să se determine un număr natural de două cifre al cărui cub are şase cifre şi se scrie numai cu cifrele `6`, `7` şi `8`.

    Exemplu. Soluţia este `92` (`92`^{3}`=778688`).

44. Să se găsească un număr `n` pentru care n^n are `n` cifre.

    Exemplu. 8^8`=16777216`.

45. Se dă de la tastatură un număr natural cu cel mult nouă cifre. Se cere să se afişeze cifrele numărului împreună cu frecvenţa lor de apariţie.

Exemplu. Pentru `n=12452` se va afişa:

0 cifre de 0	1 cifră de 5
1 cifră de 1	0 cifre de 6
2 cifre de 2	0 cifre de 7
0 cifre de 3	0 cifre de 8
1 cifră de 4	0 cifre de 9

46. Se dă un număr întreg (de tip longint). Să se afişeze cel mai mare număr obţinut prin eliminarea unei cifre din acest număr.

Exemplu. Pentru `n=6513917` se obţine numărul `653917`.

47. Să se afişeze primele `n` numere care au suma cifrelor $\leq$`m`.

Exemplu. Pentru `n=10` şi `m=4` se vor afişa `numerele 1, 2, 3, 4, 10, 11, 12, 13, 20, 21.`

48. Se citesc numere întregi până la întâlnirea numărului `0`. Să se afişeze numărul perechilor `n1` şi `n2` de numere citite consecutiv cu proprietatea că numărul cifrelor de `5` din scrierea lui `n1` este mai mare (strict) decât numărul cifrelor de `5` din scrierea lui `n2`.

Exemplu. Dacă numerele introduse sunt `182, 457, 341, 497, 5597, 1335, 15, 38, 5, 0` se va afişa valoarea `3` (perechile `457-341, 5597-1335, 15-38`).

49. Se citeşte un număr `n` şi o bază `b`. Să se verifice dacă `n` poate fi scrierea în baza `b` a unui număr.

Exemplu. Numărul `1263` poate fi considerat scris în baza `8` dar nu poate fi considerat scris în baza `6`.

50. Să se genereze toate numerele `N` de `p` cifre cu proprietatea că `N-1` şi `N+1` sunt numere prime şi în plus suma cifrelor lui `N` este tot număr prim.

Exemplu. Pentru `p=2` se vor afişa numerele `12` (`11,13,3` sunt prime) şi `30` (`29,31,3` sunt prime).

51. Pentru un întreg `n` dat, să se afişeze toate numerele naturale mai mici sau egale cu `n` a căror sumă a cifrelor este impară.

Exemplu. Pentru `n=15` se vor afişa numerele `1, 3, 5, 7, 9, 10, 12, 14`.

52. Să se genereze toate numerele prime de `n` cifre cu proprietatea că toate prefixele sale sunt de asemenea prime.

Exemplu. Pentru `n=3`, primul număr afişat va fi `113` (`1,11,113` sunt prime)

53. Să se transforme un număr din baza 10 într-o bază b, 2≤b<10.

Exemplu. $123_{10}=1111011_2$.

54. Să se transforme un număr din baza b<10 în baza 10.
Exemplu. $352_8=234_{10}$.

55. Să se transforme un număr din baza p în baza q, unde p,q ≤10.

Exemplu. $352_8=1414_5$.

56. Se citesc două numere naturale n_1 scris în baza b_1, respectiv n_2 scris în baza b_2 (b1,b2<=10). Să se afişeze max{n_1,n_2}.

Exemplu. $245_6<131_9$.

57. Să se inverseze (oglindească) un număr (care nu se termină cu cifra 0).

Exemplu. Pentru numărul 10758 se obţine numărul 85701.

58. Se citeşte un număr natural de la tastatură. Câte cifre conţine?

Exemplu. Numărul 12602 conţine 5 cifre.

59. Pentru un număr natural cu cel mult nouă cifre, să se determine numărul de cifre distincte din care se compune.

Exemplu. Numărul 13221 are 3 cifre distincte.

60. Câte cifre comune au două numere citite de la tastatură? Afişaţi care sunt aceste cifre.

Exemplu. Numerele 21348 şi 14513 au 3 cifre comune şi anume 1, 3, 4.

61. Să se afişeze toate numerele prime de trei cifre care citite invers, sunt tot numere prime.

Exemplu. Un astfel de număr este 761 (761 este prim şi 167 este prim).

62. Să se afişeze toate numerele prime ≤n care citite invers, sunt tot numere prime.

Exemplu. Pentru n=75 se vor afişa numerele 2, 3, 5, 7, 11, 13, 17, 31, 37, 71, 73.

63. Să se calculeze de câte ori se regăseşte cifra c în numerele întregi mai mici sau egale decât n dat.
Exemplu. Pentru n=79325 şi c=5 se va afişa valoarea 41763.

64. Se citeşte de la tastatură un număr natural m. Să se afişeze cifra de rang k dat (rangul unei cifre este numărul său de ordine, numerotând cifrele de la dreapta la stânga începând cu 0).

Exemplu. Cifra de rang 2 din numărul 1234567 este 5.

65. Să se listeze toate numerele ≤n, a căror sumă a cifrelor este divizibilă prin 5.

Exemplu. Pentru n=25 se vor afişa numerele 5, 14, 19, 23.

66. Scrieţi un program care afişează numerele naturale mai mici sau egale cu o valoare **n** dată, cu proprietatea că suma cifrelor lor este un număr prim.

Exemplu. Pentru **n=25** se vor afişa valorile **2, 3, 5, 7, 11, 23**.

67. Dintre numerele naturale mai mici sau egale cu o valoare **n** dată, să se afişeze acelea care sunt divizibile cu suma cifrelor lor.

Exemplu. Pentru **n=25** se vor afişa valorile **1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 18, 20, 21, 24**.

68. Un număr se numeşte palindrom dacă citit invers este acelaşi număr. Să se verifice dacă un număr citit de la tastatură este palindrom sau nu.

Exemplu. Numărul **12321** este palindrom.

69. Se citeşte un număr natural **n** de la tastatură. Să se calculeze şi afişeze restul şi câtul împărţirii lui **n** la suma cifrelor lui **n**.

Exemplu. Pentru **n=894** se obţine câtul **42** şi restul **12**.

70. Se numeşte număr "bine ordonat" descrescător/crescător un număr natural, cu proprietatea că cifrele sale citite de la stânga la dreapta sunt în ordine descrescătoare/crescătoare. Scrieţi un algoritm care verifică dacă un număr natural **x** dat este "bine ordonat" descrescător sau crescător, afişând un mesaj corespunzător.

Exemplu. Numărul **5310** este bine ordonat descrescător iar **146** este bine ordonat crescător.

71. Să se verifice dacă un număr **n** este palindrom în baza **16**. Numărul este introdus în baza **10**.
Exemplu. Numărul **111281** este palindrom în baza **16**. ($111281_{10}=1B2B1_{16}$)

72. Fiind dat un număr în baza **10**, să se verifice dacă numărul poate fi palindrom într-o bază **b** ($2 \le b \le 36$).

Exemplu. Numărul **42** (scris în baza **10**) este palindrom în bazele **4**, **13** şi **20** ($42_{10}=222_4=33_{13}=22_{20}$).

73. Se citeşte un număr natural **n** de la tastatură. Scrieţi un program care afişează numărul de apariţii ale cifrei **1** din reprezentarea numărului **n** în baza **2**.

Exemplu. În scrierea numărului **547** în baza **2** există **4** cifre de **1**.

74. Să se determine ultima cifră a sumei $1^n+2^n+...+n^n$, pentru **n** număr întreg citit de la tastatură. Nu se va calcula efectiv suma.

Exemplu. Pentru **n=29** se obţine cifra **5**.

75. Care este cifra de pe poziţia **k**, dacă se aşează unul după altul numerele **1, 2,**

Exemplu. Cifra de pe poziţia **20** este **1** (**1 2 3 4 5 6 7 8 9 10 11 12 13 14 15**). Cifra de pe poziţia **2131** este **6** (ultima cifră din numărul **746**).

76. Se citesc două numere întregi de la tastatură. Se cere să se tipărească pozițiile binare care reține valori diferite.

Exemplu. Numerele **53** și **4698** diferă pe pozițiile binare **0**, **1**, **2**, **3**, **5**, **6**, **9** și **12** ($53_{10}=110101_2$ iar $4698_{10}=1001001011010_2$)

77. Fiind date două numere naturale **a** și **b** a căror reprezentare binară ocupă **16** biți, spunem că **a** este șeful lui **b** dacă pentru orice poziție binară **0** a lui **a** poziția corespunzătoare a lui **b** este **0**. Se citește **n**, număr natural care se poate reprezenta pe **16** poziții binare. Se cere să se tipărească numărul șefilor săi.

Exemplu. Numărul **n=325** are un număr de **4096** de șefi.

78. Se citește un număr natural **n** și o bază **b** (**2<=b<=10**). Să se tipărească numărul maxim rezultat prin eliminarea unor cifre ale lui **n**, astfel încât acesta să poată fi considerat scris în baza **b**.

Exemplu. Pentru **n=475281** și **b=6** numărul obținut este **4521**.

79. Scrieți un program care rotește întregul **n** la dreapta cu **p** poziții.

Exemplu. Numărul **12345** rotit la dreapta cu **2** poziții este **45123**.

80. Scrieți un program care inversează cei **n** biți din reprezentarea în baza **2** a numărului **x**, care încep de la poziția **p** (numărând de la dreapta spre stânga începând cu **0**), lăsându-i pe ceilalți neschimbați.

Exemplu. Pentru n=1234 = 10011010010_2, **p=2** obținem $00101100110_2 = 358_{10}$.

81. Se citesc de la tastatură **n** numere naturale cu cel mult nouă cifre fiecare. Să se afișeze unul dintre aceste numere care are cele mai multe cifre distincte în scrierea sa. Dacă există mai multe numere cu același număr maxim de cifre distincte se va afișa cel mai mic dintre ele.

Exemplu. Dacă **n=5** iar numerele introduse sunt **5**, **999**, **34**, **11**, **23** atunci se va afișa numărul **23**.

82. Să se scrie un program care calculează „cifra de control" a unui număr întreg efectuând suma cifrelor sale, apoi suma cifrelor acestei sume etc. până se obține o sumă formată dintr-o singură cifră.

Exemplu. Cifra de control a numărului **1971** este **9** (**1971->18->9**).

83. Se citesc mai multe caractere până la întâlnirea caracterului „*". Expresia formată din înlănțuirea caracterelor citite reprezintă un număr zecimal periodic. Să se determine două numere întregi prime între ele a căror raport este chiar numărul dat.

Exemplu. Fracția **12.2(123)** este egală cu **40667/3330**.

84. Se dă un număr natural **k** (**k<=32000**). Se cere să se afişeze toate numerele pătrate perfecte mai mici strict decât **k** care sunt formate din cifre distincte.

Exemplu. Pentru **k=200** se vor afişa numerele **1,4,9,16,25,36,49,64,81, 169,196.**

85. Să se scrie un program care determină primele **n** cifre din seria fracţiei subunitare $\frac{k}{m} = 0.c_1c_2...c_n$ pentru **k** şi **m** (**k<m**) numere naturale citite de la tastatură.

Exemplu. Pentru **k=5, m=23,** şi **n=5** se va afişa **0.21739.**

86. Să se afişeze toate numerele de **n** cifre egale cu de **k** ori produsul cifrelor. Numerele **n** şi **k** (**1≤n≤9, 1≤k≤1000**) se citesc de la tastatură.

Exemplu. Pentru **n=2** şi **k=5** singurul număr care îndeplineşte condiţiile problemei este **175** (**5·(1·7·5)=175**).

87. Pentru un număr **q** (**1<q<10**) să se scrie toate numerele mai mici decât o valoare **x** dată care, scrise în baza **q**, folosesc numai cifrele **0** şi **1**.

Exemplu. Pentru **q=5** şi **x=30** se afişează numerele **1, 5, 25, 26, 30.**

88. Se citeşte un număr natural de cel mult **9** cifre şi să se afişeze o piramidă formată din cifrele acestui număr: pe prima linie cifra (sau cifrele) din mijloc etc., pe ultima linie fiind scris întregul număr. Se va asigura aspectul de triunghi isoscel plin al figurii.

Exemplu. Dacă este dat numărul **237855**, se afişează:

```
  78
 3785
237855
```

89. Se consideră că într-un oraş există **n** (**n≤100**) abonaţi telefonici ale căror numere de telefon sunt formate din **6** cifre, prima cifră fiind obligatoriu nenulă. Presupunem că regia RomTelecom a organizat un concurs al cărui premiu va fi oferit acelui abonat al său, al cărui număr de telefon respectă următoarele condiţii:

    se află în lista celor **n** abonaţi

    conţine doar cifre pare

    între numerele care respectă primele două condiţii va fi ales numărul care este maxim conform ordinii numerelor naturale.

Datele de intrare se citesc de la tastatură şi afişarea numărului câştigător se va face pe ecran. În cazul în care nu va exista un astfel de număr se afişează „Nu există abonat câştigător".

Exemplu. Pentru **n=4** şi lista cu numerele: **221242, 24923, 480024, 228888,** se va afişa **480024.**

90. Se introduc de la tastatură numere întregi formate din minim două cifre, până la întâlnirea numărului `0`. Să se afişeze pe ecran câte numere care au cifra unităţilor mai mică decât cifra zecilor există.

Exemplu. Dacă se introduce şirul `25, 653, 2965, 211, 154, 1256, 0` se va afişa valoarea `3`.

91. Se spune că `n` este *deosebit* dacă există un număr natural `m` astfel încât `n=m+S(m)`, unde `S(m)` este suma cifrelor lui `m`. Să se scrie un program care verifică dacă un număr `A` dat este deosebit.

Exemplu. `1235` este deosebit (`1235=1225+10`).

92. Scrieţi un program care calculează numărul de cifre obţinut prin alăturarea numerelor `1, 2, ..., n`.

Exemplu. Pentru `n=15` se obţin `21` de cifre.

93. Să se afişeze toate numerele mai mici decât `100000` egale cu suma factorialelor cifrelor componente.
Exemplu. Un astfel de număr este `145` (`145=1!+4!+5!`),

94. Se citeşte un număr întreg care va fi stocat într-o variabilă de tip longint. Să se afişeze un mesaj corespunzător dacă numărul are aspect de "deal" sau de "vale".

Exemplu. Numărul `2465320` are aspect de deal (urcă până la cifra `6` apoi coboară). Numărul `52108` are aspect de vale (coboară la cifra `0` apoi urcă).

95. Se citesc de la tastatură un număr real `x` şi un număr natural `n` (`n≤16`). Să se afişeze numărul `x` cu `n` zecimale pe ecran în formă de piramidă ca mai jos. Pentru `x=14.63` şi `n=4` se va afişa

```
      6
       3
   4    0
 1        0
```

96. Să se scrie un program care transformă un număr natural `n` din baza `b` (`b>1`) în baza b^k, `k>0`.

Exemplu. $10011101_2 = 235_8$

Indicaţii. Numărul în baza `b` este împărţit în grupe de câte `k` cifre pornind de la dreapta la stânga. Fiecare grupă este apoi transformată individual în baza `10`, valoarea obţinută fiind egală cu valoarea grupei respective în baza b^k. Numărul în baza b^k este format din cifrele obţinute din transformarea fiecărei grupe. Dacă numărul nu are un număr de cifre multiplu de `k` se va completa cu `0` în faţa numărului. Pentru exemplul de mai sus: `(010) (011) (101)=>235`.

97. Sistemul zecimal de numeraţie se bazează pe cifrele {0,1,...,9} şi numerele 1, 10, 100, 1000, Cu ajutorul cifrelor şi numerelor respective putem reprezenta orice număr natural N în forma

$$N=b_nb_{n-1}...b_1b_0 = b_n10^n + b_{n-1}10^{n-1}+...+b_210+b_0\ 1.$$

Exemplu. $538_{10}=5\cdot100+3\cdot10+8\cdot1$.

Sistemul de numeraţie Fibonacci se bazează pe cifrele {0,1} şi pe şirul Fibonacii $a_1=1,\ a_2=2,...,a_i=a_{i-1}+a_{i-2}$ ($i\geq3$). Într-un mod similar orice număr natural M poate fi reprezentat unic sub forma

$$M=(c_nc_{n-1}...c_2c_1)_F = c_n\cdot a_n + \quad + c_4\cdot5 + c_3\cdot3 + c_2\cdot2 + c_1\cdot1$$

Unde a_i sunt numerele lui Fibonacci.

$$M = 10101001_F$$

$$= 1\cdot34 + 0\cdot21 + 1\cdot13 + 0\cdot8 + 1\cdot5 + 0\cdot3 + 0\cdot2 + 1\cdot1 = 53_{10}$$

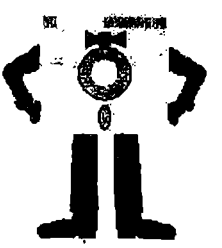
Să se scrie un program care:

a) transformă un număr întreg pozitiv din sistemul zecimal în sistemul Fibonacci;

b) transformă un număr din sistemul Fibonacci în sistemul zecimal.

98. Să se scrie un program care transformă un număr natural n din baza b^k, $1<b<10$, $k>0$, în baza b.

Exemplu. $537_8=101\ 011\ 111_2$.



2.3. *Generări de şiruri*

99. Să se afişeze primii n termeni ai următoarelor şiruri:

a) 1, 2, 4, 8, ...

b) 1, 2, 1, 1, 2, 3, 4, 3, 2, 1, 1, 2, 3, 4, 5, 6, 5, 4, ...

c) 1, 21, 123, 4321, 12345, ...

100. Să se precizeze valoarea elementului de pe poziţia k dată a şirului 1, 2, 2, 3, 3, 3, 4, 4, 4, 4, fără a citi sau construi şirul efectiv.

Exemplu. Al 35-lea termen al şirului este 8.

101. Să se determine al n-lea termen al şirului:

1, 1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5,

fără a folosi tipuri structurate de date.

Exemplu. Al 55-lea termen al şirului este 10.

102. Să se genereze primele n numere din şirul:

1, 1, 2, 2, 1, 2, 3, 3, 3, 1, 2, 3, 4, 4, 4, 4, ...

103. Să se calculeze $\sqrt{x}$ unde $x \in R$. Se va folosi metoda Newton. Această metodă constă din următoarele etape:

pornind cu $a_0=1$ se generează următorul şir de numere reale

$$a_n = \frac{a_{n-1} + \frac{x}{a_{n-1}}}{2}$$

când diferenţa dintre a_n şi a_{n-1} este foarte mică (mai mică decât o valoare $\varepsilon>0$ dată), procesul de generare se încheie;

la sfârşit, a_n, reprezintă rădăcina pătrată a lui x.

104. Să se calculeze produsul primilor n termeni ai şirului lui Fibonacci definit astfel:

$$F_1 = 1 \qquad F_2=1$$

$$F_n = F_{n-1} + F_{n-2} \text{ pentru } n\geq 3.$$

Exemplu. Pentru n=6 se obţine produsul 240.

105. Să se genereze primii n termeni ai şirului a_n definiţi astfel:

$a_1=3$

$a_n=3a_{n-1}+2\,(a_{n-2}+a_{n-3}+...+a_1+1)$.

Să se găsească o modalitate eficientă de calcul, fără utilizarea nici unui tip structurat de date.

Exemplu. $a_5=571$.

106. Se citesc coeficienţii a, b, c ai unei ecuaţii de gradul II şi un număr natural n. Fără a rezolva ecuaţia să se calculeze expresia $S_n = x_1^n + x_2^n$ unde x_1 şi x_2 sunt soluţiile ecuaţiei.

Exemplu. Pentru n=10 şi ecuaţia $3x^2+2x-7=0$ se obţine suma 610.8101069.

107. Să se tipărească termenii şirului definit prin relaţia de recurenţă:

a) $a_{n+1} = 1 + a_n^2 \quad (\forall)\ n \geq 1, \quad a_0 = 1$

Exemplu. Pentru n=5 şirul generat va fi (1, 2, 5, 26, 677, 458330).

b) $x_n = 2 \cdot x_{n-1} + x_{n-2} \quad (\forall)\ n \geq 2\,,\ x_0 = 0 \quad x_1 = 1$

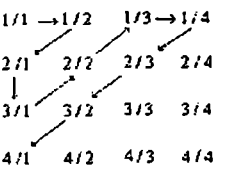
Exemplu. Pentru n=6 se va genera şirul (0, 1, 2, 5, 12, 29, 70).

108. Se consideră şirul 1, 2, 3, 4, 5, 10, 20, 40, în care termenii şirului începând cu al şaselea se determină prin relaţia `a[n]=2*a[n-1]`. Pentru un număr întreg `p` dat să se determine o descompunere a lui `p` ca sumă de termeni distincţi ai şirului dat.

Exemplu. Pentru `p=55` o descompunere este `1+4+10+40`.

109. Una din cele mai cunoscute demonstraţii din matematică este demonstraţia lui Cantor a faptului că mulţimea numerelor raţionale este numărabilă.

Se consideră următoarea modalitate de reprezentare şi parcurgere/numărare a mulţimii numerelor raţionale pozitive:

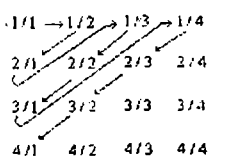


a) Fiind dat un număr natural `n` (`n<=2000000`) să se determine al `n`-lea termen din acest şir.

b) Fiind dat un element de forma `p/q` să se determine al câtelea termen este el în cadrul şirului lui Cantor.

Exemplu. Termenul al `6`-lea este `1/3`.

110. O altă modalitate de a parcurge mulţimea numerelor raţionale este următoarea



a) Fiind dat un număr natural `n` (`n<=2000000`) să se determine al `n`-lea termen din acest şir.

b) Fiind dat un element de forma `p/q` să se determine al câtelea termen este el în cadrul şirului lui Cantor.

Exemplu. Termenul al `6`-lea este `3/1`.

111. Să se genereze în ordine crescătoare primele `n` de numere din mulţimea `M`, definită astfel:

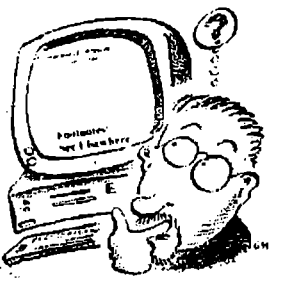
i) numărul `1` aparţine lui `M`;

ii) dacă `x` aparţine lui `M` atunci `2x+1` şi `3x+1` aparţin de asemenea lui `M`;

iii) `M` nu conţine alte elemente.

Exemplu. Primele `10` numere ale şirului sunt `1, 3, 5, 7, 9, 10, 11, 13, 15`.

2.4. Calcule de sume şi produse



112. Să se calculeze suma s=2+4+6+...+(2n), pentru un n dat de la tastatură.

Exemplu. Dacă n=56 rezultă S=3192.

113. Calculaţi suma:

$$S = \frac{1}{1 \times 2} + \frac{1}{2 \times 3} + \frac{1}{3 \times 4} + \quad + \frac{1}{n \times (n+1)}.$$

Exemplu. Pentru n=5 se obţine S=0.8333.

114. Pentru un număr întreg n dat, să se calculeze fiecare dintre expresiile de mai jos:

a) $S_1 = 1^2 + 4^2 + 7^2 + \ldots\ldots + (3n-2)^2$

b) $S_2 = 2 - 4 + 6 + \ldots\ldots + (-1)^{n+1} \quad (2n)$.

Exemplu. n=5 => S_1=335 şi S_2=6.

115. Să se calculeze suma $S = 1 - 2 + 3 - 4 + \quad \pm n$, pentru n dat de la tastatură.

Exemplu. Dacă n=10 se obţine suma -5.

116. Să se calculeze suma numerelor naturale impare mai mici sau egale decât o valoare n dată.

Exemplu. Pentru n=10 se obţine suma 25.

117. Calculaţi produsul

$$P = \left(1 - \frac{1}{2^2}\right)\left(1 - \frac{1}{3^2}\right). \ldots \left(1 - \frac{1}{n^2}\right).$$

Exemplu. Pentru n=5, P=0.6, iar pentru n=10, P=0.55.

118. Să se verifice pentru un n citit de la tastatură, egalitatea:

$$C_n^0 + C_n^1 + C_n^2 + \quad + C_n^n = 2^n$$

unde C_n^k sunt coeficienţii binomiali definiţi astfel:

$$C_n^k = \frac{n!}{k!\,(n-k)!} \qquad (x! = 1 \quad 2 \quad 3 \qquad x).$$

Exemplu. Pentru n=5 avem: 1+5+10+10+5+1=32.

119. Se citeşte un şir de numere întregi până la întâlnirea numărului 0. Să se calculeze media aritmetică a numerelor din şir.

Exemplu. Dacă se introduc numerele 56, 23, 4, 98, 14, 22, 102, 0 media afişată este 45.571429.

120. Pentru n şi k citite de la tastatură (n>=6, k>=3) să se verifice egalitatea:

$$C_{n-3}^{k} + 3C_{n-3}^{k-1} + 3C_{n-3}^{k-2} + C_{n-3}^{k-3} = C_{n}^{k}.$$

Exemplu. Pentru n=10 şi k=5 se obţine în ambele părţi ale egalităţii valoarea 252.

121. Să se scrie un program care citeşte pe x real şi pe n număr întreg, calculează valoarea polinomului:

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \quad + (-1)^n \frac{x^{2n+1}}{(2n+1)!}$$

şi scrie rezultatul calculului.

Exemplu. Dacă x=2 şi n=10 se obţine suma 0.90929742.

122. Să se calculeze produsul a două numere naturale prin adunări repetate.

Exemplu. 3*9=27 (3+3+3+3+3+3+3+3+3 sau 9+9+9).

123. Fie n valori întregi. Să se calculeze suma algebrică a valorilor de rang impar (primul, al treilea, al cincilea, ...) şi produsul tuturor valorilor diferite de zero.

Exemplu. Dacă se introduc următoarele 10 numere 2, -5, 47, 0, 32, 12, -6, 14, 0, -3 se va obţine suma 75 şi produsul -45480960.

124. Fie n valori de tip întreg. Să se calculeze raportul dintre suma elementelor strict pozitive şi produsul tuturor celor n valori.

Exemplu. Pentru n=5 şi numerele 9, -2, 7, 4, -3 se găseşte raportul 0.01322751.

125. Să se scrie un program care citeşte pe x real şi pe n număr întreg, calculează valoarea polinomului:

$$1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots + (-1)^n \frac{x^{2n}}{(2n)!}$$

şi scrie rezultatul calculului.

Exemplu. Dacă x=2 şi n=5 valoarea polinomului este -0.4161552028.

126. Să se calculeze:

$$S=1+1\cdot2+1\cdot2\cdot3+\ldots+1\cdot2\cdot3\cdot\ldots\cdot n.$$

Exemplu. Pentru n=7 suma obţinută este 5913.

127. Să se scrie un program care citeşte x (real) şi calculează valoarea expresiei:

$$\frac{(x-2)(x-4)(x-8)\ldots(x-64)}{(x-1)(x-3)(x-7)\ldots(x-63)}.$$

Se presupune că x este diferit de 1, 3, ..., 63.

Exemplu. Pentru x=1.5 se obţine valoarea 2.22288182.

128. Să se calculeze:

$$S=1-1\cdot2+1\cdot2\cdot3-\ldots\pm1\cdot2\cdot3\cdot\ldots\cdot n.$$

Exemplu. Pentru n=7 => S=4421.

129. Efectuaţi împărţirea a două numere întregi, fără a utiliza operatori de împărţire ci doar scăderi repetate.

Exemplu. 85/7=12 rest 1 (se efectuează 12 operaţii de scădere).

130. Să se scrie un program care citeşte a>0 (real), iar apoi determină şi scrie pe ecran valoarea lui n cu proprietatea

$$1+\frac{1}{2}+\frac{1}{3}+\quad+\frac{1}{n}>a.$$

Exemplu. Dacă a=2.5 atunei n=7, iar dacă a=5, n=83.

131. Se citeşte n număr natural. Să se calculeze expresia de mai jos:

$$E=1+\frac{1}{2^2}+\frac{1}{3^2}+\quad+\frac{1}{n^2}$$

Exemplu. Dacă n=5 => E=1.46361111.

132. Se citesc 3 numere naturale n, p şi k, apoi un şir de numere naturale. Câte dintre acestea, împărţite la p dau restul k.

Exemplu. În şirul de numere 12, 56, 54, 22, 9, 2, 45, 78, 99, 57 există 4 numere care împărţite la 5 dau restul 2.

133. Scrieţi un program care să calculeze valoarea unui polinom într-un punct dat. Polinomul va fi dat prin coeficienţii săi. Nu se vor utiliza tablouri.

Exemplu. Valoarea polinomului $x^5+3x^4+7x^3-9x^2+2x-3$ pe punctul x=2 este 101.

134. "*Înmulţirea á la russe*" este cunoscută încă din timpul Egiptului antic şi este o adunare cam ciudată la prima vedere.

Se scriu deînmulţitul şi înmulţitorul unul sub altul, formând sub fiecare câte o coloană, conform regulii: se împarte numărul de sub deînmulţit la 2, ignorând fracţiile, apoi se înmulţeşte cu 2 numărul de sub înmulţitor. Se aplică regula, până când numărul de sub deînmulţit este 1. În final, adunăm toate numerele

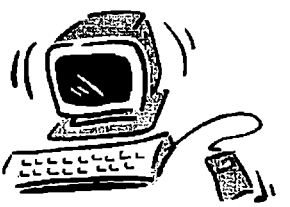
din coloana înmulțitorului care corespund, pe linie unor numere impare în coloana deînmulțitului.

Exemplu. 45 × 19 = 855

45	19	19
22	38	-
11	76	76
5	152	152
2	304	-
1	608	608
		855

Să se scrie un program care realizează înmulțirea a două numere folosind acest algoritm.

2.5. Probleme diverse



135. Să se determine toate numerele de forma $a^2 + b^3$, cu $1 \le a \le 5$ și $1 \le b \le 5$.

Exemplu. Un număr de forma dată este 68 ($68=2^2+4^3$).

136. Pentru mulțimile A={1,2,…,m} și B={1,2,…,n}, cu m și n citite de la tastatură, să se afișeze elementele mulțimii

$$A \times B = \{(a,b) \mid a \in A,\ b \in B\}.$$

Exemplu. Pentru m=3 și n=2 AxB={(1,1),(1,2),(2,1),(2,2),(3,1),(3,2)}.

137. Să se determine toate numerele întregi de trei cifre $\overline{abc}$ cu proprietatea că

$$\overline{abc} = a^3 + b^3 + c^3$$

Exemplu. Un astfel de număr este 371 ($371=3^3+7^3+1^3$).

138. Să se afișeze toate puterile mai mici decât n ale unui număr întreg k dat.

Exemplu. Pentru n=5 și k=100 se afișează numerele 1, 5, 25.

139. Se citește numărul real x. Să se determine numerele întregi p și q astfel încât:

$$p \le x < q$$

q-p minim.

Nu se va folosi nici o funcție predefinită.

Exemplu. Pentru x=12.56 se obține p=12 și q=13.

140. Se dă un număr real pozitiv `x` şi un număr întreg `n` (`n>6`). Să se calculeze x^n folosind mai puţin de `n-1` înmulţiri.

Exemplu. $2^5 = (x^2)^2x$ adică 3 înmulţiri.

141. Se citesc `n` numere întregi de la tastatură. Să se afişeze suma primelor `m` numere negative.

Exemplu. Pentru `n=16, m=5` şi numerele `1,6,-2,4,-3,7,9,10,-7,-9,11,-1, 19,-8,-20,41` se obţine suma `-22`.

142. Să se tabeleze funcţia $f(x) = \sqrt{\sin x^2 + 1}$, cu pasul `0.01`, pentru valori în intervalul `[0,1]`.

143. Să se deplaseze o literă pe ecran, folosind cele patru taste direcţionale. (Ecranul are 25 linii şi 80 coloane. Tastele au următoarele coduri: sus 72, jos 80, stânga 75, dreapta 77).

144. Izotopul radioactiv de plutoniu 135 are perioada de înjumătăţire de 26 minute, adică după 26 minute doar jumătate din izotop va mai rămâne. După alte 26 minute doar jumătate din această jumătate ş.a.m.d. Scrieţi un program care va determina cât timp îi va lua plutoniului 135 să scadă până la punctul în care un procent specificat din cantitatea iniţială a mai rămas.

Exemplu. Pentru un procent dat de `0.5%` timpul necesar este de 208 minute.

145. Introduceţi de la tastatură o rată medie de creştere a populaţiei Terrei şi calculaţi după câţi ani faţă de 1992 se va dubla populaţia. (În 1992 Terra avea o populaţie de 5480 milioane de locuitori).

Exemplu. Dacă rata de creştere este de `0.05` (adică `5%` pe an) atunci populaţia se dublează în 15 ani.

146. În povestea "Jack şi vrejul de fasole", Jack trebuia să ajungă în Ţara Uriaşului, situată la `x` metrii deasupra pământului, urcând cu vrejul de fasole care avea proprietatea miraculoasă de a creşte în fiecare minut cu $\frac{1}{2}, \frac{1}{3}, \frac{1}{4}$, etc. din înălţimea lui anterioară (înălţimea iniţială a vrejului este `1 m`). În câte minute ajunge Jack în Ţara Uriaşului?

Exemplu. Dacă `x=300` Jack ajunge în vârf după 598 de minute.

147. La o societate sunt `n` (`n<32`) acţionari. Cel mai important acţionar deţine din capitalul societăţii un procent concretizat prin `c` milioane lei (`c>1`). Următorul ca importanţă deţine jumătate din procentul deţinut de primul, următorul jumătate din procentul deţinut de al doilea, etc. Realizaţi un program care tipăreşte suma deţinută de ultimul acţionar.

Exemplu. Dacă sunt 5 acţionari iar suma deţinută de primul acţionar este de 130 de milioane atunci ultimul acţionar deţine `8.125` milioane (adică 8 milioane şi ...).

148. Cunoscându-se numele şi înălţimea a `n` elevi, să se afişeze numele elevului cu înălţimea cea mai mică şi numele elevului cu înălţimea cea mai mare.

Observaţie. Nu există doi elevi cu aceeaşi înălţime. Nu se vor folosi tipuri de date structurate.

Exemplu. Dacă elevii sunt `A, B, C, D, E` şi au înălţimile `150, 165, 140, 180, 170` atunci elevul cel mai înalt este `D` iar cel mai mic este `C`.

149. Să se afişeze numerele pătrate perfecte formate din nouă cifre distincte.

Exemplu. Primul astfel de număr este `1026753849`.

150. Să se scrie un program care determină toate reprezentările posibile ale unui număr natural ca sumă de numere naturale consecutive.

Exemplu. Pentru `n=50` avem `50=8+9+10+11+12` şi `50=11+12+13+14`.

151. Se consideră două mulţimi: `V` conţinând valori binare `0, 1`, iar cealaltă `O` conţinând operatori logici simbolizaţi prin `N (not), A (and), O (or)`. Din aceste mulţimi se obţine un rezultat logic astfel: se iniţializează rezultatul cu prima valoare din `V`. În continuare se citesc valorile din `O` până la întâlnirea caracterului `#`. Dacă se citeşte `N`, se neagă rezultatul curent, dacă se citeşte `A` sau `O`, se citeşte următoarea valoare din `V` şi se face conjuncţia, respectiv disjuncţia. Să se afişeze rezultatul obţinut în final.

Exemplu. Dacă valorile introduse în ordine sunt `1, 1, 1, 0, 1, 1` şi operatorii sunt în ordine `N, A, O, O, N, N, A, O` atunci rezultatul final va fi `1`.

152. Să se realizeze următoarea "piramidă":

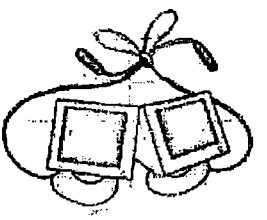
```
1
1 2
1 2 3

1 2 3      n
```

153. Să se realizeze următoarea piramidă de numere:

```
n n-1 n-2      3 2 1

3 2 1
2 1
1
```



154. Să se realizeze următoarea "piramidă" de numere:

```
          1
        1 2 3

1 2           (2n-1)
```

155. Se consideră următoarea piramidă de numere:

```
 1
 3  5
 7  9 11
13 15 17 19
```

Să se determine cu ce număr începe şi se termină linia `n`, cu `n` dat de la tastatură. Să se verifice apoi că suma termenilor de pe orice linie este cub perfect, iar pentru o linie dată `k` să se afişeze numărul al cărui cub este egal cu suma elementelor de pe linia `k`.

Exemplu. Linia `8` începe cu valoarea `57` şi se termină cu valoarea `71`. Suma termenilor de pe linia `8` este $512=4^3$.

156. Scrieţi un program care va citi un întreg şi un caracter. Programul va afişa un romb compus din caracterul citit de la tastatură şi având lăţimea specificată de întreg. Dacă întregul este un număr par, el va fi mărit cu `1`.

Exemplu. Dacă întregul este `9` şi caracterul este '`X`', rombul va arăta astfel:

```
    X
   XXX
  XXXXX
 XXXXXXX
XXXXXXXXX
 XXXXXXX
  XXXXX
   XXX
    X
```

157. Afişaţi următorul tabel de operaţii:

```
1·9 +  2 = ?
12·9 +  3 = ?
123·9 +  4 = ?

123456789·9 + 10 = ?
```

(Semnelc de întrebare se vor înlocui cu rezultatele operaţiilor respective).

158. Afişaţi următorul tabel de operaţii:

```
9·9 + 7 = ?
98·9 + 6 = ?
987·9 + 5 = ?

98765432·9 + 0 = ?
```

(Semnele de întrebare se vor înlocui cu rezultatele operaţiilor respective).

159. Se citeşte n. Se cere k astfel încât numărul C_n^k să fie maxim $\left(C_n^k = \frac{n!}{k!\,(n-k)!}\right)$.

Exemplu. Pentru n=7 numărul cerut este k=3.

160. Se citeşte o dată calendaristică sub forma z,l,a. Să se afişeze data zilei următoare.

Exemplu. Ziua următoare datei de 28.02.2002 este 01.03.2002.

161. Cunoscându-se data curentă exprimată prin trei numere reprezentând anul, luna, ziua precum şi data naşterii unei persoane, exprimată la fel, să se scrie un program care calculează vârsta persoanei respective în număr de ani împliniţi.

Exemplu. Dacă data curentă este 04.03.2002 iar persoana s-a născut pe data de 05.09.1980 atunci ea are 21 de ani împliniţi.

162. Se dau z,l,a reprezentând o dată calendaristică (coreetă). Să se afişeze câte zile mai sunt până la sfârşitul anului.

Exemplu. Dacă data calendaristică este 22.03.2002 atunci până la sfârşitul anului mai sunt 284 de zile.

163. Dându-se trei numere întregi reprezentând o dată calendaristică sub forma an, lună, zi, să se scrie un program care stabileşte a câta zi din an este aceasta.

Exemplu. Data de 22.03.2000 este a 82 zi din an, iar data de 22.03.2002 este a 81 zi din an (vezi problema 26 de la capitolul 1).

164. Dându-se data curentă prin trei numere întregi (an,lună,zi) şi cunoscându-se a câta zi a săptămânii este prima zi din anul respectiv, să se determine a câta zi din săptămână este ziua datei respective.

Exemplu. Prima zi a anului 2002 a fost a doua zi din săptămână şi deci data de 11.03.2002 a fost o zi de luni.

165. Scrieţi un program care să citească mai multe numere întregi, până la întâlnirea lui zero, şi să determine cel mai mare şi cel mai mic număr citit.

Exemplu. Dacă numerele introduse sunt 2, 9, 32, 5, -5, 8, 0 atunci maximul este 32 iar minimul -5.

166. Se citesc numere de tip întreg până când se introduce un număr de două ori unul după altul. Să se afişeze câte dintre acestea sunt pătrate perfecte. Ultimul număr introdus în şir (cel care este egal cu cel precedent) nu va fi numărat!

Exemplu. Pentru şirul 169, 144, 12, 22, 25, 6, 16, 16 se va afişa numărul 4.

167. Se citesc de la tastatură `n` numere naturale. Să se afişeze media aritmetică a elementelor pozitive din şir.

Exemplu. Pentru `n=5` şi numerele `1, -6, 9, -7, 3` media este `4.3333`.

168. Se consideră `n` valori reale. Memorând numai câte una, să se afişeze câte numere din şir sunt mai mari sau egale decât numărul întreg `a`.

Exemplu. Pentru `n=10`, `a=5` şi numerele `1, 9, 3, 6, 4, 2, 5, 7, 8, 10` se găseşte valoarea `6`.

169. Să se scrie un program care, pentru un număr natural dat `n`, determină cel mai mare număr `p` cu proprietatea că $2^p \leq n$.

Exemplu. Pentru `n=133` obţinem `p=7` ($2^7=128$, $2^8=256$).

170. Se consideră `n` valori întregi. Memorând numai câte una, să se determine valoarea cea mai mare negativă şi valoarea cea mai mică pozitivă, specificând dacă sunt sau nu unice.

Exemplu. Pentru `n=11` şi numerele `2,6,-5,9,-7,1,3,-2,55,-6,1` se vor afişa valorile `-2` şi `1`, prima fiind unică în timp ce `1` apare de `2` ori.

171. Se citesc de la tastatură pe rând mai multe numere, atâta timp cât nu s-a introdus valoarea `0` (aceasta nu face parte din şir). Să se afişeze suma elementelor pozitive şi suma elementelor negative din şir.

Exemplu. Dacă se introduce şirul `1, 2, -3, 4, -5, 6, 7, 8, -9, 10, 11, 0` se vor afişa sumele `49` şi `-17`.

172. Se citesc de la tastatură mai multe litere până la întâlnirea caracterului "`.`". Să se contorizeze numărul de vocale citite.

Exemplu. În şirul de litere '`aceastaesteopropozitiefaraspatii`' sunt `17` vocale.

173. Se citesc de la tastatură mai multe litere până la întâlnirea caracterului "`.`". Câte perechi de vocale alăturate sunt în şirul citit.

Exemplu. În şirul '`aceastaesteopropozitiefaraspatii`' sunt `5` astfel de perechi.

174. Se introduce de la tastatură un şir de numere întregi, distincte două câte două până la întâlnirea numărului zero. Se cere să se afişeze cele mai mici două elemente din şir.

Exemplu. Dacă se introduc numerele `5, 7, 9, 8, 6, 3, 4, 2, 10, 0` se vor afişa valorile `2` şi `3`.

175. Se citesc de la tastatură caractere, până întâlnirea a două caractere care sunt identice. Să se determine câte caractere sunt cifre dintre cele citite.

Exemplu. În şirul '`abd34m5j88`' sunt `5` cifre.

176. Se citesc două numere naturale n şi k şi apoi n numere reale. Să se verifice dacă, în ordinea intrării, cele n numere sunt ordonate crescător până la al k-lea număr şi apoi descrescător până la sfârşit. Se va afişa un mesaj corespunzător.

Exemplu. Pentru n=5 şi k=2 şirul 1, 10, 8, 6, 5 este ordonat după criteriul enunţat.

177. Să se rezolve în mulţimea numerelor întregi ecuaţia $x^2+y^2=k$, pentru k citit de la tastatură.

Exemplu. Pentru k=200 se obţin două soluţii: (2,14) şi respectiv (10,10).

178. Să se determine dacă două numere sunt termeni consecutivi ai şirului lui Fibonacci.

Exemplu. 55 şi 89 sunt termeni consecutivi în şirul Fibonacci.

179. Se citesc coordonatele a n puncte în plan. Câte din aceste puncte se găsesc pe dreapta de ecuaţie ax+by+c=0?

Exemplu. Dacă a=1, b=5, c=-3 atunci din punctele (-7,2), (0,0), (3,0), (1,1), (5,9) doar 2 puncte se găsesc pe dreaptă.

180. Să se reconstituie următoarea adunare cifrată:

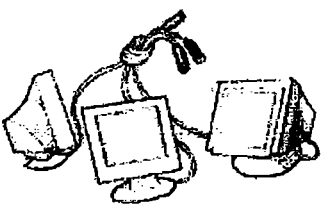
```
VINO + GINO = LANOI
```

Literelor diferite le corespund cifre diferite ale bazei 10. Cifra L are valori ≤2. Să se afişeze toate soluţiile.

Exemplu. Prima soluţie a problemei este 3874+6874=10748.

181. Să se reconstituie înmulţirea

```
   ATOM x
   ATOM
   ----
  *****
 *****
*****
*****
--------
****ATOM
```



Se ştie că fiecare literă şi fiecare steluţă reprezintă cifre de la 0 şi 9. Steluţele iniţiale ca şi litera A nu pot reprezenta cifra 0, iar la litere diferite corespund cifre diferite. Să se afişeze toate soluţiile.

Exemplu. O soluţie este ATOM=9376. Câte astfel de soluţii mai există?

182. Următoarea înmulțire criptată este o problemă care poate fi rezolvată înlocuind caracterele * cu cifre de la 0 la 9:

```
  *** x
   **
  ---
 ****
****
-----
*****
```

Observație. Prima cifră a oricărui produs parțial, a rezultatului final, a înmulțitorului sau a deînmulțitului nu poate fi zero. În plus cele 10 cifre din înmulțit, înmulțitor și din rezultat să fie distincte două câte două, cu alte cuvinte să se utilizeze toate cele 10 cifre 0, 1, ..., 9 în cele trei numere.

Exemplu. 297 × 54 = 16038.

183. *Problema maimuței.* Pe o insulă se află trei marinari și o maimuță. Marinarii au reușit să strângă o grămadă de nuci de cocos și s-au hotărât să le împartă frățește a doua zi. În timpul nopții unul din marinari s-a sculat, a împărțit grămada în trei părți egale, ascunzând una din părți, rămânând o nucă în plus, a dat-o maimuței. Până dimineața povestea s-a repetat și cu ceilalți doi marinari. Dimineața marinarii împart grămada rămasă în trei părți egale, rămânând din nou o nucă în plus care i-a revenit maimuței. Să se determine toate numerele $n \le 1000$ care permit derularea exactă a evenimentelor prezentate. Câte nuci revin în fiecare din cazuri fiecărui marinar și maimuței?

Exemplu. O soluție este următoarea: numărul inițial de nuci de cocos = 727, primul marinar ia 313 nuci, al doilea 232, al treilea 178, iar maimuța 4.

184. Să se scrie un program care calculează pătratul unui număr cu maxim 5000 de cifre, care începe cu 2 și următoarele cifre sunt toate 3.

Exemplu. 23333^2=544428889.

185. Să se scrie un program care calculează pătratul unui număr cu maxim 5000 de cifre, care începe cu 2 și următoarele cifre sunt toate 6.

Exemplu. 266^2=70756.

186. Să se scrie un program care calculează pătratul unui număr cu maxim 5000 de cifre, care începe cu 2 și următoarele cifre sunt toate 9.

Exemplu. 2999^2=8994001.

187. Să se scrie un program care calculează pătratul unui număr cu maxim 5000 de cifre, care începe cu 3 și următoarele cifre sunt toate 6.

Exemplu. 3666^2=1344395556.

188. Să se scrie un program care calculează pătratul unui număr cu maxim 5000 de cifre, care începe cu 4 şi următoarele cifre sunt toate 3.

Exemplu. $4333^2=18774889$.

189. Să se scrie un program care calculează pătratul unui număr cu maxim 5000 de cifre, care începe cu 5 şi următoarele cifre sunt toate 3.

Exemplu. $5333^2=28440889$.

190. O carte se paginează cu numere de la 1 la n ($20 \le n \le 10000$, n natural). Fiind dat n, să se afle câte cifre au fost folosite la paginare. Invers, dacă se dă numărul m de cifre pentru paginare, să se afle numărul de pagini al cărţii ($31 < m < 38894$).

Exemplu. Pentru o carte cu 100 de pagini se folosesc 192 de cifre (9 pagini cu câte 1 cifră; 90 de pagini cu câte două cifre; 1 pagină cu 3 cifre).

191. Să se scrie un program pentru generarea şi afişarea tablei lui Pitagora, până la $N \le 10$. Rezultatul se va afişa pe ecran sub forma unei matrice pătratice de ordin N.

Exemplu. Tabela lui Pitagora pentru N=10 arată astfel:

```
    |  1   2   3        j        10
----------------------------------
  1 |  1   2   3        j        10
  2 |  2   4   6      2*j        20
    |
  i |  i i*2 i*3      i*j      I*10
    |  ..........................
 10 | 10  20  30     10*j       100
```

192. Fie funcţia lui Collatz, definită astfel:

$$f(n) = \begin{cases} \frac{n}{2} & \text{daca n este par} \\ 3n + 1 & \text{daca n este impar} \end{cases}$$

folosind această funcţie se poate construi un şir de numere ce "tinde" la 1 astfel: $x_0=n$, $x_1=f(x_0)$, $x_2=f(x_1)$, ..., $x_{p-1}=f(x_{p-2})$, $x_p=f(x_{p-1})=1$.

a) Se cere să se scrie un program care determină secvenţa de lungime maximă obţinută pentru $n \le 100$.

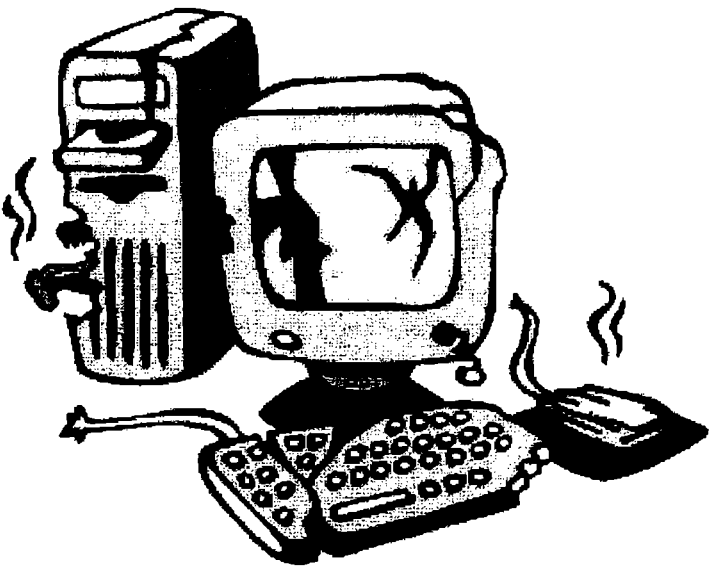
Exemplu. Secvenţa de lungime maximă se obţine pentru n=97 şi are lungimea 119.

b) Să se determine toate valorile lui `n` (`n<=100`) pentru care lungimea secvenței este `k`, cu `k` citit de la tastatură.

Exemplu. Pentru `k=10` se vor afișa valorile `12, 13, 80, 84, 85`.

c) Pentru o valoare dată a lui n să se afișeze șirul obținut ca mai sus.

Exemplu. Pentru `n=13` se obține șirul: `13, 40, 20, 10, 5, 16, 8, 4, 2, 1`.



3. Tablouri

3.1. Tablouri unidimensionale

3.1.1. Parcurgeri (sume, numărări)

1. Fiind dat un vector de numere întregi, să se determine suma elementelor pozitive din vector.

   Exemplu. Pentru vectorul (- 1, 23, 2, - 4, 5) se va afişa 30 (=23+2+5).

2. Să se afişeze doar elementele pare dintr-un vector de numere întregi.

   Exemplu. Pentru vectorul (-2,4,15,3,8,13) se va afişa -2, 4, 8.

3. Scrieţi un program care tipăreşte elementele pare aflate pe poziţii divizibile cu 3 dintr-un vector de numere întregi.

   Exemplu. Daca vectorul iniţial este (1,2,8,-4,7,9,120,53) se va afişa 8.

4. Să se afişeze elementele pozitive de rang impar dintr-un vector de numere întregi.

   Exemplu. Pentru vectorul (2,-3,-1,43,3,1,-4,8,12) se va afişa 2,3,12.

5. Se citeşte de la tastatură un vector cu n (1≤n≤100) componente de tip întreg. Se cere să se construiască şi să se afişeze un nou vector cu componentele pătrate perfecte din vectorul iniţial.

   Exemplu. Daca vectorul iniţial este (2,4,8,25,3,66) al doilea vector va fi (4,25).

6. Să se scrie un program care numără elementele pozitive, negative şi respectiv nule dintr-un vector, parcurgându-l o singur dată.

   Exemplu. În vectorul (-3,4,30,-1,0,-22,0,376) se găsesc 3 numere pozitive, 3 numere negative şi 2 numere nule.

7. Să se determine media aritmetică a elementelor pare dintr-un vector de numere întregi.

   Exemplu. Pentru vectorul (-2,4,15,3,6,13) se va afişa 2.66.

8. Să se calculeze suma elementelor negative şi de valoare pară dintr-un vector de numere întregi.

   Exemplu. Suma cerută în cazul vectorului (12,-4,9,7,-5,31,-74,515, -33,-8) este -86.

9. Fiind dat un vector `v` cu `n` elemente numere întregi, scrieți un program care calculează produsul și numărul elementelor negative aflate pe poziții impare în vector.

   Exemplu. În cazul vectorului `(12,-4,9,7,-5,31,-74,515,-33,-8)` avem `3` componente negative pe pozițiile impare iar produsul lor este `-12210`.

10. Să se afișeze poziția pe care apare primul element pozitiv dintr-un vector citit de la tastatură.

    Exemplu. În vectorul `(-5,-8,-6,-41,-9,-1,-475,-12,-4)` nu există nici un număr pozitiv, iar în vectorul `(-4,9,874,-54)` primul număr pozitiv se găsește pe poziția `2`.

11. Se dă un vector `v` cu `n` elemente numere întregi. Să se copieze într-un alt vector `u` elementele pozitive ale vectorului inițial.

    Exemplu. Dacă `v=(-5,8,-6,0,-9,1,-475,12,4)` atunci `u=(8,1,12,4)`.

12. Se consideră un vector `A` cu `m` elemente și două numere reale `P`, `Q`. Să se scrie un program care copiază într-un vector `B` toate elementele din `A` aflate în intervalul `(P,Q)` în ordinea în care apar ele în vector.

    Exemplu. Dacă `A=(12,-4,9,7,-5,31,-74,515,-33,-8)` și `P=-10`, `Q=9` atunci `B=(-4,7,-5,-8)`.

13. Se consideră un vector `A` cu `m` elemente și două numere reale `P`, `Q`. Să se scrie un program care copiază într-un vector `B` toate elementele din `A` aflate în intervalul `(P,Q)` în ordinea inversă apariției lor în vectorul `A`. Componentele vectorului `B` vor fi generate direct în această ordine!

    Exemplu. Dacă `A=(12,-4,9,7,-5,31,-74,515,-33,-8)` și `P=-10`, `Q=9` atunci `B=(-8,-5,7,-4)`.

14. Se dă un vector de numere reale. Să se determine procentul de numere din vector mai mici (strict) decât `MA` și procentul de numere din vector mai mari (strict) decât `MA`, unde am notat cu `MA` media aritmetică a componentelor din vector.

    Exemplu. Dacă vectorul este `(-2,13,34.75,125,3.5,8,61)` atunci media aritmetică este `34.75`, numerele mai mici strict decât această medie sunt `-2`, `13`, `3.5`, `8` (adică `57.14%`), iar numerele strict mai mari decât medie sunt `125`, `61` (adică `28.57%`).

15. Se consideră doi vectori cu componente reale. Se citește un număr natural `k`. Să se intercaleze în primul vector, după poziția `k`, tabloul al doilea.

    Exemplu. Dacă `x=(1,2,3,4,5,6,7)`, `y=(9,10,11)` și `k=4` dupa intercalare `x=(1,2,3,4,9,10,11,5,6,7)`.

16. Scrieţi un program care citeşte de la tastatură cele `n` numere reale ce compun vectorul `a` şi apoi cele `m` numere reale ce constituie componentele vectorului `b` şi afişează pe ecran câte dintre componentele vectorului `a` sunt strict mai mici decât toate componentele vectorului `b`.

Exemplu. Dacă `a=(4,8,1,9,5,11,3,43,6,20)` şi `b=(9,9,6,9,9,8,6,9)`, atunci numărul căutat este `4`, deoarece valorile `4,1,5` şi `3` sunt mai mici decât toate elementele lui `b`.

17. Să se calculeze şi afişeze media aritmetică şi cea armonică a componentelor unui vector `x` de numere reale. Prin medie armonică a vectorului $x=(x_1, x_2, \ldots, x_n)$ se înţelege raportul:

$$\frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \quad + \frac{1}{x_n}}.$$

Exemplu. Pentru `x=(1,8,4,21,3,56,9)` media aritmetică este `14.57` iar media armonică este `3.71`.

18. Scrieţi câte un program care să realizeze inversarea unui vector:
    a) în acelaşi vector şi fără a utiliza un vector suplimentar;
    b) într-un alt vector.

Exemplu. Dacă vectorul iniţial este `(1,2,3,4)` cel final va fi `(4,3,2,1)`.

19. Fie `v` un vector de numere întregi. Să se construiască un vector `w`, astfel încât `w[i]`= numărul de apariţii ale lui `v[i]` în vectorul `v`.

Exemplu. Pentru `v=(1,5,2,1,5,7,2,1,5)` se obţine `w=(3,3,2,3,3,1,2,3,3)`.

20. Fiind dat un vector `v` cu `n` elemente numere întregi, să se construiască alţi doi vectori: primul va conţine numai elementele pare, iar al doilea numai elementele impare ale vectorului iniţial.

Exemplu. Daca vectorul iniţial este `v=(1,64,2,5,23,9,6,11)` se vor obţine vectorii `(64,2,6)` şi respectiv `(1,5,23,9,11)`.

21. Să se calculeze componentele şirului $Y = (y_1, y_2, \ldots, y_n)$ ştiind că y_i, $i = 1,2,\ldots,n$ este media aritmetică a componentelor pozitive de rang mai mic sau egal cu `i` ale vectorului `X`, în cazul în care există componente pozitive, respectiv `-1` în caz contrar.

Exemplu. Pentru `x=(-2,-3,71,43,-3,1,-4,8,12)` se obţine `y=(-1,-1,71,57,57,38.33,38.33,30.75,27)`

22. Să se verifice dacă un vector conţine elemente în ordinea: negativ, negativ, pozitiv, negativ, negativ, pozitiv, ...

Exemplu. `(-1,-2,3,-5,-6,2,-8)` are aspectul cerut `(-1,2,-3,-4)` nu are aspectul dorit.

23. Fie dat un vector $x = (x_1, x_2, x_3, \ldots, x_n)$. Să se modifice vectorul astfel încât în final să avem:

a) $x = (x_2, x_3, \ldots, x_n, x_1)$

b) $x = (x_n, x_1, x_2, \ldots, x_{n-1})$

c) $x = (x_2, x_1, x_4, x_3, \ldots, x_n, x_{n-1})$.

24. Un tablou a cu elemente în mulțimea {0,1} se numește alternativ dacă pentru orice i, a[i] a[i + 1] = 0. Să se scrie un program care să decidă dacă un astfel de tablou citit de la tastatură este alternativ sau nu.

Exemplu. Vectorul (1,0,0,3,0,0,0,5) este alternativ însă vectorul (1,0,0,3,4,0,0,0,0,3) nu este alternativ.

25. Se dă numărul natural n>1 și vectorul $x = (x_1, x_2, \ldots, x_n)$ să se verifice dacă numerele date sunt în progresie aritmetică sau geometrică și să se dea un mesaj corespunzător.

Exemplu. Vectorul (2,4,6,8,10) este în progresie aritmetică iar vectorul (2,10,50,250,1250,6250) este în progresia geometrică.

26. Fie un vector x de numere întregi. Să se formeze un vector y de numere întregi, în care y[i] să fie restul împărțirii lui x[i] la suma cifrelor lui x[i]. Restul împărțirii se va determina folosind:

a) operatorii al limbajului;

b) scăderi repetate.

Exemplu. Pentru x=(12,5,123,85,7,33,12,8,26) se va obține y=(0,0,3,7,0,3,0,0,2).

27. Să se afișeze elementele prime ale unui șir de n numere întregi citit de la tastatură.

Exemplu. Pentru vectorul (101,10,3301,1021,5,15,7,91,37,29) se va afișa 101, 3301, 1021, 5, 7, 37, 29.

28. Memorați în primele n componente ale unui vector x de numere întregi, primele n numere prime mai mari decât 999, care citite invers, sunt tot numere prime.

Exemplu. Pentru n=5 se obțin numerele 1009, 1021, 1031, 1033, 1061.

29. Câte elemente din tabloul $x=(x_1, x_2, \ldots, x_n)$ de numere întregi au exact k divizori proprii.

Exemplu. Pentru vectorul (1,584,165,45,102,65,47,312,125,945) și k=6 se va afișa 3 (numerele 584, 165 și 102 au exact 6 divizori proprii).

30. La codificarea unui text scris cu cel mult primele `n` litere mici ale alfabetului englez, se folosesc cele mai mici `n` numere naturale care au exact 3 divizori naturali. Pentru litera '`a`' se foloseşte cel mai mic număr cu proprietatea respectivă, pentru '`b`' cel mai mic număr diferit de cel folosit pentru '`a`', şi aşa mai departe.

    a) Să se afişeze, separate prin câte un spaţiu, cele `n` numere folosite la codificare.

    b) Pentru un text dat de la tastatură, se cere să se afişeze codificarea textului, fiecare caracter fiind înlocuit cu numărul corespunzător şi orice două coduri consecutive vor fi separate printr-un spaţiu.

    Exemplu. Pentru `n=5` şi textul '`abeaab`' se va afişa a) `4, 9, 25, 49, 121`; b) `4 9 121 4 4 9`.

31. Să se calculeze `cmmdc(`$x_1, x_2, x_3, \ldots, x_n$`)`.

    Exemplu. `cmmdc(2940,882,70,182)` este `14`.

32. Câte numere perfecte există în tabloul `x=(`$x_1, x_2, \ldots, x_n$`)` de numere întregi citite de la tastatură. Un număr este perfect dacă este egal cu suma divizorilor săi mai mici decât el.

    Exemplu. Pentru vectorul `(10,5,6,12,8128,5,0,28)` se va afişa `3` (deoarece `6`, `8128` şi `28` sunt numere perfecte).

33. Să se afişeze valorile întregi `p` şi `q`, prime între ele astfel încât $\frac{p}{q} = \frac{a_1}{b_1} + \frac{a_2}{b_2} + \quad + \frac{a_n}{b_n}$, numerele întregi a_i, b_i, $i = \overline{1, n}$, dându-se de la tastatură.

    Exemplu. Pentru `a=(1,5,3,6,11)` şi `b=(2,4,2,7,4)` se obţine `p=48` şi `q=7`.

34. Câte elemente dintr-un vector de numere întregi sunt prime cu un număr dat.

    Exemplu. Pentru vectorul `x=(12,15,254,525,56,125,500,63,48,912)` şi numărul `4` se obţin `4` numere `(15,525,125,63)`.

35. Se dă un număr întreg `n`. Să se determine numerele `n1` şi `n2` obţinute din cifrele de ordin impar, respectiv par ale numărului `n`, în ordinea apariţiei lor în numărul `n` şi apoi afişează suma celor două numere obţinute.

    Exemplu. Dacă `n=32457` se obţin numerele `347` şi `25` iar suma lor este `372`.

36. Cu câte zerouri se termină produsul elementelor x_i, $i = \overline{1, n}$, ale unui vector de numere întregi, fără a calcula produsul.

    Exemplu. Pentru vectorul `x=(12,15,254,525,56,125,500,105,48,912)` se va afişa `10`.

37. Scrieţi un program care elimină toate elementele nule dintr-un vector de numere întregi.

Exemplu. Dacă vectorul iniţial este `x=(10,0,0,3,0,5,0,9)` se va obţine `x=(10,3,5,9)`.

38. Se dă un vector `v` cu `n` elemente numere întregi. Fără a folosi un vector auxiliar, să se mute la sfârşitul lui `v` elementele sale nule, păstrând ordinea celorlaltor elemente.

Exemplu. Dacă iniţial `v=(2,3,0,9,0,0,8)`, în final va rezulta `v=(2,3,9,8,0,0,0)`.

39. Se consideră un şir de `n` numere reale. Să se scrie un program care elimină din şir valorile ce se află în afara intervalului `[a,b]`.

Exemplu. Dacă vectorul iniţial este `(2,8,4,6,9,10,3,5,2,10)` iar `a=3` şi `b=7` atunci vectorul rezultat va fi `(4,6,3,5)`.

40. Să se scrie un program care inserează între oricare două elemente ale vectorului `x` de numere reale, media aritmetică a vecinilor săi.

Exemplu. Dacă iniţial `x=(2,5,10,3,7)` după prelucrare `x=(2,3.5,5,7.5,10,6.5,3,5,7)`.

41. Fiind dat un vector cu `n` numere reale, să se determine valoarea expresiei `e` în fiecare din cazurile:

a) `e = x[1] + x[2] + + x[n]`

Exemplu. Pentru `x=(10,2,31,335,12,45,26)` => `e=461`

b) `e = x[1] + x[3] + + x[...]`

Exemplu. Pentru `x=(10,2,31,335,12,45,26)` => `e=79`

c) `e = x[1] * x[2] * x[3] * * x[n]`

Exemplu. Pentru `x=(10,2,3,5,12,4)` => `e=14400`

d) `e = x[1] * x[3] * . . .x[...]`

Exemplu. Pentru `x=(10,2,3,5,12,4)` => `e=360`

e) `e`= suma cuburilor componentelor negative din vector;

Exemplu. Pentru `x=(10,2,-3,-5,12,45,-26)` => `e=-17728`.

f) `e`=media aritmetică a componentelor impare din vector.

Exemplu. Pentru `x=(10,2,3,5,12,45,26)` => `e=17.6666`.

42. Fie $a_1, a_2, \ldots, a_n$, `n` numere reale. Calculaţi suma

$$S = a_1^2 - a_2^2 + a_3^2 - \quad + a_n^2.$$

Exemplu. Pentru vectorul `a=(45,8,1,98,6)` se obţine suma `s=-7606`.

43. Fie doi vectori `x` şi `y` cu `n` componente numere întregi fiecare, să se calculeze expresia `e` dată prin:

a) $e = (x_1 + y_1) \times (x_2 + y_2) \times \quad \times (x_n + y_n)$

Exemplu. Pentru `x=(1,2,3)` şi `y=(4,5,6)` se obţine `e=315`.

b) $e = x_1y_1 + x_2y_2 + \quad + x_ny_n$

Exemplu. Pentru `x=(1,2,3)` şi `y=(4,5,6)` se obţine `e=32`.

c) $e = x_1^2y_1 + x_2^2y_2 + \quad + x_n^2y_n$

Exemplu. Pentru `x=(1,10,2,3)` şi `y=(4,5,6,8)` se obţine `e=600`.

d) $e = \min(x_1, y_n) + \min(x_2, y_{n-1}) + \quad + \min(x_n, y_1)$

Exemplu. Pentru `x=(1,10,2,3)` şi `y=(4,5,6,8)` se obţine `e=12`.

44. Să se calculeze raportul

$$e = \frac{\sum_{i=2k} x_i}{\sum_{j=2k+1} x_j}.$$

Exemplu. Pentru `x=(2,7,4,54,2,77,12)` => `e = 6.9`.

45. Dându-se `n` numere întregi să se decidă dacă există un număr majoritar în această secvenţă. Un număr este majoritar dacă numărul său de apariţii în vector este mai mare decât `n/2`.

Exemplu. În vectorul `(1,5,3,1,1,1,4)` numărul `1` este element majoritar.

46. Fie un şir de `n` numere reale $a_1, a_2, \ldots, a_n$. Să se calculeze suma:

$$S = \frac{a_1}{1} + \frac{a_2}{2} + \quad + \frac{a_n}{n}.$$

Exemplu. Pentru vectorul `a=(45,8,1,98,6)` se obţine suma `s=75.03333`.

47. Să se determine numărul de componente din vectorul $C=(C_1, C_2, C_3, \ldots, C_n)$ aflate între două limite date `a` şi `b` şi media aritmetică a componentelor aflate în afara intervalului.

Exemplu. Dacă `C=(2,8,4,6,9,10,3,5,2,10)` iar `a=3` şi `b=7` se vor afişa valorile `4` şi respectiv `6.8333`.

48. Se dau doi vectori `A`, `B` de lungimi egale cu elemente numere întregi. Să se scrie un program care să calculeze media aritmetică a tuturor elementelor din `A` pentru care componentele respective din `B` sunt numere negative.

Exemplu. Dacă `A=(1,2,3,4,5,6,7)` iar `B=(2,5,-3,6,-9,-5,10)` se obţine media `4.6666`.

49. Dându-se un vector x (citit de la tastatură) să se calculeze vectorul y știind că

$$y_1 = \frac{(x_1 + x_2)}{2} \qquad y_2 = \frac{(x_1 + x_2 + x_3)}{3}$$

$$y_n = \frac{(x_{n-1} + x_n)}{2} \qquad y_{n-1} = \frac{(x_{n-2} + x_{n-1} + x_n)}{3}$$

iar y_k este media numerelor $x_{k-2}, x_{k-1}, x_k, x_{k+1}, x_{k+2}$ pentru $k = \overline{3, n-2}$.

Exemplu. Pentru x=(1,5,7,2,8,12,3,90,6,22) => y=(3,4.33,4.6, 6.8,6.4,23,23.8,26.6,39.33,14).

50. Fie vectorul $u = (u_1, u_2, \ldots, u_m)$ cu

$$u_i = \frac{1}{3i-2}\frac{1}{3i-1}\frac{1}{3i}.$$

Să se construiască vectorul $v = (v_1, v_2, \ldots, v_n)$ știind că $v_1 = S_{20}$, $v_i = S_{20+5i}$, $i = 1,2,\ldots,n$ unde $S_k = u_1 + u_2 + \quad + u_k$.

Observație. Se va citi de la tastatură doar valoarea lui n.

Exemplu. Pentru n=5 se va obține vectorul v=(0.1787513,0.1787764, 0.1787818,0.1787853,0.1787877)

51. Se citesc n numere naturale. Aceste numere se împart în grupe astfel încât în cadrul fiecărei grupe toate numerele au același număr de cifre 1 în reprezentarea în baza 2. Se cere să se afișeze mediile aritmetice a numerelor din fiecare grupă.

Exemplu. Pentru n=10 și numerele 92,60,47,16,52,45,65,7,8,87 se obțin următoarele medii aritmetice: 12.65,29.5,65.66,67.

52. Să se calculeze ultima cifră a numărului x[1]+x[2]+...+x[n], unde x este un vector cu n componente numere naturale citite de la tastatură.

Exemplu. Pentru n=10 și vectorul x=(92,60,47,16,52,43,65,7,8,87) se obține cifra 7.

53. Să se calculeze ultima cifră a numărului $x[1]^{x[2]*x[3]*\ldots*x[n]}$, unde x este un vector cu n componente numere naturale citite de la tastatură.

Exemplu. Ultima cifră a expresiei $53^{74*683*37*83}$ este 9.

54. Să se scrie un programul pentru calculul coeficienților de corelație a două variabile aleatoare discrete X și X. unde:

X ia valorile: X[1], X[2], X[n].

Y ia valorile: Y[1], Y[2], Y[n].

Observație: Fie RXY coeficientul de corelație a lui X și Y. El este definit de formula:

$$RXY = \frac{\sum_{i=1}^{N} (X[i] - MX)(Y[i] - MY)}{\sqrt{\left(\sum_{i=1}^{N} (X[i] - MX)^2\right)\left(\sum_{i=1}^{N} (Y[i] - MY)^2\right)}}$$

unde MX și MY sunt mediile aritmetice a componentelor din X și Y.

Exemplu. Coeficientul de corelație pentru variabilele aleatoare x=(4,8,9,2,4) și y=(8,7,3,74,9) este RXY=-0.6884783.

55. Din șirul de numere 1, 2, 3, ..., n se elimină pe rând câte un număr formând un nou șir. Un număr eliminat este introdus într-un al doilea șir, inițial vid, exact în mijlocul acestuia, dacă acest lucru este posibil, în caz contrar se adaugă la sfârșitul șirului. Se cere:

   a) dându-se ordinea în care se elimină numerele să se afișeze șirul final construit ca mai sus;

   b) dându-se ordinea numerelor din șirul final, să se determine ordinea în care au fost eliminate numerele din șirul inițial.

Exemplu. Dacă n=7 și numerele sunt eliminate astfel 1,3,7,4,2,6,5 atunci șirul final va fi 1,7,2,5,3,4,6 și invers.

56. Într-o grădină zoologică sunt n colivii cu papagali. O maimuță jucăușă trece prin fața coliviilor și deschide toate ușile acestora. Apoi, se întoarce și închide ușile din două în două (începând cu prima colivie). Jocul său se reia mergând din trei în trei colivii, închizând ușile deschise și deschizându-le pe cele închise. Apoi maimuța merge din patru în patru, din cinci în cinci, ș.a.m.d. în final din n în n colivii, începând de fiecare dată cu prima colivie. Să se afișeze coliviile rămase deschise.

Exemplu. Pentru un număr de 10 colivii vor rămâne deschise coliviile 2,5,10.

57. Se consideră n aruncări de zar (generate aleator). Se cere:

   a) să se determine de câte ori apare fiecare față și la a câta aruncare apare prima dată.

   b) să se determine toate perechile de forma (Fi, K_i) cu proprietatea că $Fi+K_i$ este un număr par, unde Fi este numărul feței, iar K_i numărul de apariții a feței F_i.

Exemplu. La o rulare a acestui program s-au obținut următoarele rezultate (n a fost generat și el aleator): fața 1 de 4704 ori; fața 2 de 4603 ori; fața 3 de 4596 ori; fața 4 de 4717 ori; fața 5 de 4655 ori; fața 6 de 4714 ori. Perechile care îndeplinesc condiția de la punctul b) sunt așadar (5,4655) și (6,4714).

58. Se citesc de la tastatură cele n elemente ale unui şir dat de numere întregi. Să se afişeze toate perechile de elemente ale şirului (nu neapărat consecutive) cu proprietatea că ambele elemente ale perechii au aceeaşi sumă a cifrelor.

Exemplu. Pentru vectorul `(12,9,54,32,222,49,34,6,1,91)` se vor afişa perechile `(9,54)` şi `(222,6)`.

59. Să se afişeze toate tripletele de numere crescătoare de pe poziţii consecutive din vectorul x de numere reale.

Exemplu. Pentru `x=(2,9,41,61,6,24,84,1,21)` se vor afişa tripletele `(2,9,41)`, `(9,41,61)` şi `(6,24,84)`.

60. Fiind dat un vector de numere întregi, să se afişeze de câte ori găsim două elemente aflate pe poziţii consecutive în vector care să fie egale între ele.

Exemplu. Pentru vectorul `(1,2,2,3,4,4,4,5,6)` se va afişa valoarea 3.

61. Fie un vector x de numere întregi. Să se afişeze toate tripletele de numere consecutive din x, în care al treilea număr este restul împărţirii primului la al doilea.

Exemplu. Pentru vectorul `(21,7,15,4,3,1,45,7,3,1)` tripletele cerute sunt `(15,4,3)`, `(4,3,1)`, `(45,7,3)` şi `(7,3,1)`.

62. Fie un vector x de numere întregi. Să se afişeze toate tripletele de numere consecutive din x, în care al treilea este media aritmetică dintre primul şi al doilea.

Exemplu. Pentru vectorul `(2,9,3,6,10,3,7,5,6,5)` tripletele cerute sunt `(9,3,6)`, `(3,7,5)` şi `(7,5,6)`.

63. Fie un vector x de numere întregi. Să se afişeze toate tripletele de numere consecutive din x, în care al treilea este media geometrică dintre primul şi al doilea.

Exemplu. Pentru vectorul `(1,3,2,5,125,25,4,8,33,8,7)` singurul triplet este `(5,125,25)`.

64. Fie un vector x de numere întregi. Să se afişeze toate perechile de numere consecutive din x, cu proprietatea că al doilea este egal cu suma cifrelor primului număr.

Exemplu. Pentru vectorul `(124,7,12,3,4,4)` perechile cerute sunt `(124,7)`, `(12,3)`, `(4,4)`.

65. Fie un vector x de numere întregi. Să se afişeze toate tripletele de numere consecutive din x, cu proprietatea că ele pot reprezenta laturile unui triunghi.

Exemplu. Singurul triplet din vectorul `(12,45,10,5,9,4,51)` este `(10,5,9)`.

66. Fie un vector x de numere întregi. Să se afişeze toate perechile de numere consecutive din x, cu proprietatea că al doilea reprezintă restul împărţirii primului număr la suma cifrelor sale.

Exemplu. Pentru vectorul (124,5,12,44,4,9) se afişează perechile (124,5) şi (44,4).

67. Fie un vector x de numere întregi. Să se afişeze toate perechile de numere consecutive din x, cu proprietatea că al doilea reprezintă numărul de apariţii ale cifrei 3 în pătratul primului.

Exemplu. Pentru vectorul (361,2,5,1156,4,0) perechile afişate vor fi (361,2), (1156,4), (4,0).

68. Fie un vector x de numere întregi. Să se afişeze toate perechile de numere (x[i], x[i + 1]) cu proprietatea că x[i] > x[i + 1] şi suma cifrelor lui x[i] este mai mică decât suma cifrelor lui x[i + 1].

Exemplu. Pentru vectorul x=(40,76,17,21,55,92,59,33,52,37) se vor afişa perechile (92,59) şi (52,37).

69. Fie un vector x de numere întregi. Să se afişeze toate tripletele de numere consecutive din x, cu proprietatea că ele pot reprezenta laturile unui triunghi dreptunghic.

Exemplu. Tripletul (3,4,5) poate reprezenta laturile unui triunghi dreptunghic.

70. Fie un vector x de numere întregi. Să se afişeze toate tripletele de numere consecutive din x, cu proprietatea că al ele pot reprezenta laturile unui triunghi isoscel.

Exemplu. Tripletul (12,20,12) este un triplet care îndeplineşte condiţiile problemei.

71. Fie un vector x de numere întregi. Să se afişeze toate perechile de numere consecutive din x, cu proprietatea că al doilea reprezintă pătratul numărului de apariţii ale cifrei 1 în primul.

Exemplu. Pentru vectorul (121,4,89,7,0,111,9) se vor afişa perechile (121,4), (7,0), (111,9).

72. Fie un vector x de numere reale. Să se afişeze soluţiile tuturor ecuaţiilor de gradul II: $a x^2 + b x + c = 0$, unde a, b şi c sunt toate tripletele de numere consecutive din x, în care $b^2 - 4ac = 0$.

Exemplu. Pentru x=(1,-2,1,12,36,5) se va afişa 1, -6. (soluţiile ecuaţiilor $x^2-2x+1=0$ respectiv $x^2+12x+36=0$)

73. Fie un vector x de numere reale. Să se afişeze soluţiile tuturor ecuaţiilor de gradul II: $a x^2 + b x + c = 0$, unde a, b şi c sunt toate tripletele de numere consecutive din x, în care $b^2 - 4ac > 0$.

Exemplu. Pentru vectorul x=(41,81,53,60,58,1,81,65,67,53) se vor afişa următoarele perechi de soluţii: (-0.95,-0.02) corespunzătoare ecuaţiei $60x^2+58x+1=0$ şi (-80.19,-0.81) care corespund ecuaţiei $x^2+81x+65=0$.

74. Fie un vector x de numere întregi. Să se afişeze soluţiile tuturor ecuaţiilor de gradul I, ax+b=0, unde a şi b sunt toate perechile de numere consecutive din vectorul x, în care b este divizibil cu prin a. Dacă un număr este soluţia mai multor astfel de ecuaţii se va afişa o singură dată.

Exemplu. Dacă x=(7,3,8,2,140,20,5,28,7,1) se găsesc soluţiile –4 şi –7.

75. Fie un vector x de numere întregi. Să se formeze un vector y de numere întregi, în care y[i] să fie reprezentarea în baza 2 a numărului x[i].

Exemplu. x=(12,8,3,14,6,21) => y=(1100,1000,11,1110,110, 10101).

76. Să se scrie un program care verifică dacă un vector x cu n componente numere întregi este o permutare a mulţimii (1,2,3,...,n).

Exemplu. Vectorul (2,6,1,4,3,5) reprezintă o permutare a mulţimii (1,2,3,4,5,6).

77. Fie x un vector care conţine o permutare a elementelor 1,2,...,n. Se defineşte un al doilea vector xcod astfel: xcod[i] este numărul de valori din {x[1],x[2],...,x[i – 1]} mai mici decât x[i]. Să se scrie un program care pentru un x dat construieşte tabloul corespunzător xcod.

Exemplu. Pentru permutarea (5,3,4,1,6,2) vectorul xcod este (0,0,1,0,4,1).

78. De la tastatură se introduce gradul n al unui polinom şi coeficienţii întregi ai acestuia. Să se afişeze rădăcinile întregi distincte ale acestui polinom.

Exemplu. Polinomul $x^5+x^4-29x^3-69x^2-84x+180$ are ca rădăcini întregi pe 1, –5 şi 6 (acestea se află printre divizorii întregi ai termenului liber).

79. O funcţie $f \{1,2,\ldots,m\} \to \{1,2,\ldots,n\}$ se poate reprezenta în calculator printr-un vector $F = (F_1, F_2, \ldots, F_m)$ cu m componente unde $F_i = f(i)$. Să se verifice dacă funcţia f, dată prin vectorul F este injectivă.

Exemplu. Funcţia corespunzătoare vectorului (3,8,2,3,6,9) nu este injectivă (f(1)=f(4)), iar funcţia corespunzătoare vectorului (1,5,2,8,14) este injectivă.

80. Dându-se funcțiile $f: \{1,2,\ldots,m\} \to \{1,2,\ldots,n\}$ și $g: \{1,2,\ldots,n\} \to \{1,2,\ldots,p\}$ reprezentate prin vectorii F și G, $F_i = f(i), G_i = g(i)$, să se determine compunerea celor două funcții date, adică $g \circ f$.

Exemplu. Dacă m=6, n=7, p=8, f=(1,3,2,4,6,3), g=(2,6,1,4,5,8,7) atunci g ∘ f=(2,1,6,4,8,1).

81. Se dă un șir X de numere întregi. Să se determine mediile v_i, i = 1,2,...,k. Prin v_i s-a notat media numerelor pozitive cuprinse între al (i-1)-lea și al i-lea număr negativ din X, dacă există numere pozitive, respectiv 0 în caz contrar.

Exemplu. X=(5,-2,3,9,-6,2,6,9,14,-1,3,-4,-7) => V=(5,6, 7.75,3,0).

82. Determinați valoarea unui polinom într-un punct. Coeficienții (reali) ai polinomului precum și punctul în care se calculează valoarea polinomului se citesc de la tastatură.

Exemplu. Valoarea polinomului $P(x)=x^5+2x^3+3x+7$. memorat folosind vectorul a=(7,3,0,2,0,1) pe punctul x=1.5 este 25.84375.

83. Să se transforme un număr din baza 10 în baza 16, cifrele numărului în baza 16 memorându-se într-un vector.

Exemplu. Pentru n=11224 se obține vectorul cu resturile împărțirilor (2,11,13,8), deci numărul în baza 16 este 2BD8.

84. Să se calculeze 2^n, pentru valori mari ale lui n.

Exemplu. 2^{100}=1267650600228229401496703205376 adică 31 de cifre.

85. Considerând că cifrele a două numere sunt memorate în doi vectori x, respectiv y să se calculeze:

a) suma celor două numere;

b) diferența celor două numere;

c) produsul celor două numere.

Exemplu. Numerele 235 și 126 se pot memora astfel: (5,3,2), respectiv (6,2,1). Această scriere inversă a cifrelor poate ușura foarte mult calculele. Se va obține suma 361, diferența 109 (atenție se va scădea cel mai mic număr din cel mai mare) și produsul 29610.

86. Să se determine numărul de cifre egale cu cifra c (dată de la tastatură) în cadrul numărului n! cu n<=500.

Exemplu. Cifra 2 apare de 3 ori în numărul 13! (13!=6227020800).

87. Se consideră doi vectori binari $x=(x_1,x_2,\ldots,x_n)$ şi $y=(y_1,y_2,\ldots,y_n)$ reprezentând scrierea în baza doi a două numere întregi. Să se calculeze şi afişeze suma şi diferenţa binară acestor numere. (Nu se va folosi scrierea în baza 10 a numerelor!)

Exemplu. În cazul vectorilor $x=(1,1,1,0,1,0,1,0)$, $y=(1,1,0,0)$ suma binară este $x+y=(1,1,1,1,0,1,1,0)$, iar diferenţa binară este $x-y=(1,1,0,1,1,1,1,0)$.

88. Numerele de la 1 la n sunt aşezate în ordine crescătoare pe circumferinţa unui cerc, astfel încât n ajunge lângă 1. Începând cu numărul s, se elimină din cerc numerele din k în k, după fiecare eliminare cercul strângându-se. Care va fi numărul ce va rămâne ultimul?

Exemplu. Dacă vectorul iniţial este (1,2,3,4,5,6,7,8,9,10) iar k=3, s=1 se vor elimina pe rând numerele 3, 6, 9, 2, 7, 1, 8, 5, 10 şi rămâne numărul 4.

89. Aceeaşi problemă ca la problema anterioară dar numerele nu se elimină, ci doar se marchează, până unul din ele va fi marcat de 2 ori. Câte numere au rămas nemarcate?

Exemplu. Dacă vectorul iniţial este (1,2,3,4,5,6,7,8,9,10), pasul este k=4 iar s=1, se vor marca pe rând numerele 4, 8, 2, 6, 10 deci vor rămâne 5 numere nemarcate.

90. Se citeşte de la tastatură un număr natural n. Se cere să se afişeze următoarele valori:

$$\tau(n) = (e_1 + 1)(e_2 + 1)\ldots(e_m + 1)$$

$$\varphi(n) = \frac{n}{p_1 p_2 \cdots p_m}(p_1 - 1)(p_2 - 1)\ldots(p_m - 1)$$

$$\sigma(n) = \frac{p_1^{e_1+1} - 1}{p_1 - 1}\,\frac{p_2^{e_2+1} - 1}{p_2 - 1}\cdots\frac{p_m^{e_m+1} - 1}{p_m - 1}$$

unde $n = p_1^{e_1} p_2^{e_2} \ldots p_m^{e_m}$ este descompunerea în factori primi a lui n.

Exemplu. Pentru n=4712 găsim τ=16, φ=2160, σ=9600.

91. Fiind dat vectorul x cu n componente să se calculeze sumele:

$$S_1 = x_1 + x_2 + \quad + x_n$$

$$S_2 = x_1x_2 + x_1x_3 + \quad + x_1x_n + x_2x_3 + \quad + x_{n-1}x_n$$

$$S_n = x_1x_2x_3\ldots x_n.$$

Indicaţie. Se observă ca sumele de mai sus sunt date de relaţiile lui Vieta în polinomul $(x-x_1)(x-x_2)\ldots(x-x_n)$

Exemplu. Pentru n=4, x_1=4, x_2=5, x_3=-1, x_4=2 se obţin sumele S_1=10, S_2=27, S_3=2, S_4=-40.

92. Se dă un şir de n cifre $a=(a_i)$, $1 \le i \le n \le 50$. Să se scrie un program care calculează:

$$S=a_1a_2\ldots a_n + a_2a_3\ldots a_{n-1}a_na_1 + \quad + a_ja_{j+1}\ldots a_na_1a_2\ldots a_{j-1} + \quad + a_na_1a_2\ldots a_{n-2}a_{n-1}.$$

Se vor afişa şi termenii sumei.

Exemplu. Pentru $a=(5,6,7,8,9)$ obţinem $S=56789+67895+78956+89567+95678=388885$.

93. Pentru un număr natural $n>0$, dat de la tastatură să se genereze într-un vector x elementele mulţimii $A_n = \{x = 2^i3^j5^k,\ i,j,k \ge 0 \text{ si } x \le n\}$. Vectorul x trebuie obţinut ordonat crescător (nu se va sorta ulterior).

Exemplu. Pentru $n=20$ vectorul este $x=(2,3,4,5,6,8,9,10,12,15,16,18,20)$.

94. Memorând coeficienţii a două polinoame sub forma a doi vectori se cere să se determine

a) suma celor două polinoame

b) produsul polinoamelor

Exemplu. Pentru polinoamele $P(x)=x^3+2x+5$ şi $Q(x)=x^3+5x+7$ se obţine

$P(x)+Q(x)=2x^3+7x+12$ şi $P(x)*Q(x)=x^6+7x^4+12x^3+10x^2+39x+35$.

95. Determinaţi câtul împărţirii polinomului $P(x) = a_0x^n + a_1x^{n-1} + \quad + a_{n-1}x + a_n$ prin $x - \alpha$, folosind schema lui Horner. Reamintim că schema lui Horner constă în următorul procedeu:

Se pun coeficienţii polinomului pe prima linie a unui tabel, urmaţi de valoarea α;

Se copiază (coboară) primul coeficient pe linia a doua;

În continuare, linia a doua se completează recurent: pentru a obţine o nouă valoare, se înmulţeşte cu α ultima valoare determinată şi se adună cu numărul aflat deasupra poziţiei curente (pe prima linie).

Ultima valoare obţinută este restul împărţirii celor două polinoame, iar valorile anterioare, în ordine, reprezintă coeficienţii câtului.

Exemplu. Pentru $P(x)=2x^3-3x^2+x-2$ şi $\alpha=4$ se obţine câtul $2x^2+5x+21$ şi restul 82:

2	-3	1	-2	$\alpha=4$
2	5	21	82	

96. Se citeşte de la tastatură un număr natural n. Se cere să se descompună în factori primi acest număr folosind doi vectori: unul pentru memorarea factorilor primi, iar al doilea cu exponenţii corespunzători acestor factori.

Exemplu. Pentru n=4712 cei doi vectori vor fi p=(2,19,31), e=(3,1,1), ceea ce reprezintă următoarea descompunere în factori primi: $2^3*19^1*31^1$.

97. Se consideră şirul 1, 2,1, 3,2,1, 4,2,2, 5,4,3,2,1, 6,2,2,3,3,3,.... obţinut din şirul numerelor naturale prin înlocuirea fiecărui număr natural printr-un grup de numere, după următoarele reguli: numărul prim p este înlocuit prin numerele p, p-1, p-2,..., 3, 2, 1, iar numărul compus k este înlocuit prin k urmat de toţi divizorii săi proprii, un divizor d repetându-se de d ori. Dându-se numărul natural n se cere să se tipărească primele n numere din şirul dat.

Exemplu. Dacă n=10 şirul generat va fi (1,2,1,3,2,1,4,2,2,5).

98. Pe un traseu circular se găsesc n depozite de morcovi. Se dau distanţele dintre două depozite consecutive (1->2, 2->3, ..., n-1->n, n->1). Fiecare dintre depozite conţine o anumită cantitate de morcovi (posibil zero). Numărul total de morcovi din cele n depozite ajung unui iepuraş exact cât poată parcurge tot traseul o singură dată. Se consideră că pentru parcurgerea unui km iepuraşul are nevoie să mănânce un morcov pentru a avea energie suficientă. Iepuraşul poate mânca oricâţi morcovi. Să se determine de la ce depozit şi în ce direcţie va pleca iepuraşul, astfel încât el să poată reveni la locul de plecare.

Exemplu. Pentru n=5, distanţele (12,3,61,10,4) şi numărul de morcovi din depozite (0,20,40,0,30) se va pleca de la depozitul 3 spre depozitul 2, sau de la depozitul 5 spre depozitul 1.

99. Considerăm şirul de numere naturale:

1; 1,1; 2,1; 1,2, 1,1; 1,1, 1,2, 2,1; 3,1, 2,2, 1,1;

definit astfel:

primul grup (subşir) este format dintr-un singur 1;

plecând de la ultimul grup cunoscut, următorul grup se construieşte prin indicarea perechilor de forma (factor de repetiţie, număr care se repetă) referitoare la conţinutul grupului de la care s-a plecat. Astfel, din grupul 1,1,1,2,2,1 se obţine 3,1,2,2,1,1 deoarece apar în ordine 3 de 1, 2 de 2 şi 1 de 1.

Să se genereze primele n componente ale şirului definit astfel.

100. Când dorim să listăm o broşură cu mai multe foi îndoite la mijloc ordinea de listare este alta decât de obicei. O astfel de broşură are patru pagini pe foaie, cu două pe faţă şi două pe spate. Când aşezăm toate aceste foi în ordine şi îndoim teancul în două, paginile apar în ordinea corectă, ca într-o carte

obișnuită. Problema cere să se scrie un program care, citind numărul de pagini ce trebuiesc tipărite, generează ordinea de tipărire. Eventualele pagini goale ce trebuie să fie adăugate se vor adăuga la sfârșitul cărții.

Exemplu. Pentru `n=10` paginația este următoarea:

foaia 1: fața: –, 1; spatele: 2, –;

foaia 2: fața: 10, 3; spatele: 4, 9

foaia 3: fața: 8, 5, spatele: 6, 7.

101. Un număr de `n` copii așezați pe un cerc joacă următorul joc. Pe rând fiecare din copii spune un număr după următoarea regulă:

primul copil spune `1`, următorii spunând numărul dinainte `+1`.

dacă unui copil îi vine rândul să spună un număr divizibil cu `5`, atunci el spune "`PAS`" și sensul jocului se schimbă, iar dacă numărul este divizibil cu `7` sau are ultima cifră `7` sensul jocului se schimbă deasemenea.

Exemplu. `n=4`

copilul 1 spune 1
copilul 2 spune 2
copilul 3 spune 3
copilul 4 spune 4
copilul 1 spune PAS – se schimbă sensul
copilul 4 spune 6
copilul 3 spune 7 – se schimbă sensul
copilul 4 spune 8
copilul 1 spune 9
copilul 2 spune PAS – se schimbă sensul
copilul 1 spune 11
copilul 4 spune 12
copilul 3 spune 13
copilul 2 spune 14 – se schimbă sensul
copilul 3 spune PAS – se schimbă sensul
copilul 2 spune 16
copilul 1 spune 17 – se schimbă sensul
copilul 2 spune 18
copilul 3 spune 19
ș.a.m.d

Să se scrie un program, care citește de la tastatură numărul de copii care vor să joace acest joc și numărul de pași ai jocului și afișează copiii care au spus PAS. Dacă un copil spune PAS de mai multe ori, el se va afișa o singură dată.

Exemplu. Pentru `n=10` (copiii) și numărul de pași `np=50` se va afișa valoarea 2 (doi copii nu au spus PAS și anume 2 și 8).

102. De la tastatură se citesc cele `N` elemente ale unui șir de numere. Să se scrie un program care interschimbă două secvențe de lungimi `L1` și `L2` care încep din pozițiile `p1`, respectiv `p2`. Elementele care nu sunt cuprinse în cele două secvențe pot fi mutate astfel încât interschimbarea să se poată face. Nu se vor folosi vectori auxiliari.

Exemplu. Pentru `n=9`; șirul `1,2,3,4,5,6,7,8,9`, `P1=2`, `L1=2`, `p2=6`, `L2=3`, rezultatul este `1,6,7,8,4,5,2,3,9`.

103. *Generarea permutărilor.* Să se scrie un program care generează toate permutările mulțimii $\{1,2,\ldots,n\}$ folosind ordinea lexicografică a acestora.

Indicații. Se pleacă de la cea mai mică permutare în sens lexicografic adică $(1,2,\ldots,n)$. La un anumit pas, având construită permutarea $p=(p_1,p_2,\ldots,p_n)$, pentru a determina următoarea permutare se determină acel indice i pentru care

$$p_i<p_{i+1} \qquad p_{i+1}>p_{i+2}>\ldots>p_n.$$

Următoarea permutare se obține prin înlocuirea lui p_i cu cel mai mic dintre elementele $p_{i+1}, p_{i+2}, \ldots, p_n$ care este mai mare decât p_i, urmată de inversarea ordinii ultimelor $n-i$ elemente, încât ele să apară în ordine crescătoare. Dacă nu există nici un indice i ca mai sus înseamnă că s-a obținut permutarea cea mai mare, $\{n,n-1,\ldots,1\}$ și algoritmul se încheie.

Exemplu. Pentru $n=4$ permutările sunt generate în următoarea ordine: $(1,2,3,4)$, $(1,2,4,3)$, $(1,3,2,4)$, $(1,3,4,2)$, $(1,4,2,3)$, $(1,4,3,2)$, $(2,1,3,4)$, $(2,1,4,3)$, $(2,3,1,4)$, $(2,3,4,1)$, $(2,4,1,3)$ etc.

104. *Generarea combinărilor.* Se cere să se genereze toate cele C_n^m submulțimi de m elemente ale mulțimii $\{1,2,3,\ldots,n\}$, cu proprietatea că oricare două astfel de submulțimi diferă prin natura elementelor (nu și prin ordinea lor). Se va folosi generarea în ordine lexicografică a acestor submulțimi.

Indicații. Se vor genera deci toți vectorii $v=(v_1,v_2,\ldots,v_m)$ care satisfac relația

$$1<=v_1<v_2<\ldots<v_m<=n.$$

Se pleacă de la vectorul cel mai mic în sens lexicografic adică $(1,2,\ldots,m)$. La un anumit pas, fiind dat vectorul $v=(v_1,v_2,\ldots,v_m)$, vectorul care-i urmează în sens lexicografic se determină astfel: se determină cel mai mare indice i satisfăcând relațiile

$$v_i<n-m+i \qquad v_{i+1}=n-m+1,\ldots,v_{m-1}=n-1,\ v_m=n.$$

Următorul vector va fi $v=(v_1,v_2,\ldots,v_{i-1},v_i+1,v_i+2,\ldots,v_i+n-i+1)$. Dacă nu există nici un astfel de indice i înseamnă că $v=(n-m+1,n-m+2,\ldots,n)$ adică au fost generate toate combinările de m elemente ale mulțimii $\{1,2,\ldots,n\}$.

Exemplu. Pentru $n=5$ și $m=3$ combinările se vor genera în următoarea ordine: $(1,2,3)$ $(1,2,4)$ $(1,2,5)$ $(1,3,4)$ $(1,3,5)$ $(1,4,5)$ $(2,3,4)$ $(2,3,5)$ $(2,4,5)$ $(3,4,5)$.

105. *Generarea combinărilor.* Se cere să se genereze toate cele C_n^m submulțimi de m elemente ale mulțimii $\{1,2,3,\ldots,n\}$, cu proprietatea că oricare două astfel de submulțimi diferă prin natura elementelor (nu și prin ordinea lor). De această dată se va folosi reprezentarea submulțimilor unei mulțimi prin

vectori caracteristici. Astfel o submulțime S a mulțimii A={1,2,...,n} va fi reprezentată prin vectorul caracteristic v=(v_1,v_2,...,v_n) unde

$$v_i = \begin{cases} 1 & \text{daca } i \in S \\ 0 & \text{daca } i \notin S \end{cases}$$

Vectorii v se vor genera în ordine lexicografică, plecând de la vectorul v=(0,0,...,0,1,1,...,1), în care primele n-m componente sunt nule iar următoarele m componente sunt egale cu 1.

Exemplu. Pentru n=5 și m=3 combinările se generează în acest caz în ordinea (3,4,5) (2,4,5) (2,3,5) (2,3,4) (1,4,5) (1,3,5) (1,3,4) (1,2,5) (1,2,4) (1,2,3).

106. *Generarea aranjamentelor.* Se cere să se genereze toate cele A_n^m submulțimi de m elemente ale mulțimii {1,2,3,...,n}, cu proprietatea că oricare două astfel de submulțimi diferă fie prin natura elementelor fie prin ordinea lor acestora. Generarea aranjementelor se va face folosind ordinea lexicografică.

Indicație. Se pleacă de la aranjamentul {1,2,...,m}, cea mai mică în sens lexicografic. Având un aranjament a=(a_1,a_2,...,a_m) pentru a determina succesorul său, în ordine lexicografică, căutăm cel mai mare indice i cu proprietatea că a_i poate fi mărit (adică măcar una din valorile a_i+1, a_i+2,...,n este disponibilă). Dacă există un astfel de indice, aranjamentul următor este obținut din a prin înlocuirea elementelor a_i, a_{i+1}, ... a_m cu cele mai mici numere disponibile în ordine crescătoare.

Exemplu. Pentru n=5 și m=4 aranjările se vor afișa în următoarea ordine:
(1,2,3,4) (1,2,3,5) (1,2,4,3) (1,2,4,5) (1,2,5,3)
(1,2,5,4) (1,3,2,4) (1,3,2,5) (1,3,4,2) (1,3,4,5)
(1,3,5,2) etc.

107. Un număr de n (n≤100) celule sensibile sunt așezate în cerc, fiecare comunicând cu cei doi vecini ai săi. O celulă se poate găsi în două stări: "excitată" sau "liniștită". Dacă o celulă este excitată la un moment dat atunci ea emite un semnal care ajunge după un timp scurt (să zicem o milisecundă) la cele două celule vecine cu ea. Fiecare celulă este excitată atunci și numai atunci când la ea ajunge un semnal de la una dintre celulele vecine; dacă semnalele ajung deodată din amândouă părțile, atunci ele se anulează și celula nu se mai excită. Dându-se o configurație inițială de celule excitate și liniștite, se cere să se determine configurația după n pași.

Exemplu: Dacă configurația inițială este 100110101001101010001101 atunci configurația după 10 pași va fi: 101010100000000001010101.

3.1.2. Maxime şi minime

108. Se dau doi vectori `x`, `y` cu `n` componente (`1≤n≤100`). Primul conţine nume de elevi, iar al doilea notele la informatică. Se cere să se afişeze numele elevilor cu nota cea mai mică. Dacă mai mulţi elevi au această notă minimă se vor afişa toţi.

Exemplu. Dacă `x=(Ionescu, Iliescu, Antonescu, Popescu, Pătrulescu, Piscupescu)` iar `y=(6,5,9,10,5,7)` se va afişa: `Iliescu, Pătrulescu`.

109. Să se realizeze programul pentru determinarea elementului maxim şi a tuturor poziţiilor unde apare el într-un vector `A` de numere întregi cu dimensiunea `N`.

Exemplu. Elementul maxim din vectorul `(2,7,3,57,2,50,57,57,4,9)` este `57` şi se găseşte pe poziţiile `4, 7, 8`.

110. Să se determine elementele cu număr maxim de divizori şi elementele cu număr minim de divizori dintr-un vector `x` de numere întregi.

Exemplu. Pentru vectorul `x=(59,748,15,665,12,4,7,99,68,74)` componentele cu număr minim de divizori (2) sunt `59` şi `7` iar cu număr maxim de divizori este numărul `748` (12 divizori).

111. Se dau coordonatele a `n` puncte în plan (x_i, y_i), $i = \overline{1,n}$. Să se afişeze lungimea celui mai lung segment şi coordonatele capetelor acestui segment.

Exemplu. Dacă punctele sunt `(1,8), (4,5), (4,2), (4,9), (3,9)` atunci segmentul de lungime maximă va avea capetele de coordonate `(4,2)` respectiv `(3,9)` şi va avea lungimea `7.07`.

112. Dându-se un vector cu `n` componente numere întregi, să se determine cele mai mici trei valori pozitive şi cele mai mari trei valori negative. Nu se va folosi nici o metodă de sortare şi nu se vor modifica valorile componentelor din vector.

Exemplu. Dacă vectorul dat este `(-37,87,84,-65,-18,-90,93,-76,91, -3,-6,-14,41,-27,61,-49,-45,51,62, 38)` atunci cele mai mici trei valori pozitive sunt: `38, 41, 51` iar cele mai mari trei valori negative: `-14, -6, -3`.

113. Se dă un vector conţinând litere. Să se afişeze frecvenţa apariţiei fiecărei litere în vector (nu se face distincţie între literele mari şi cele mici) şi literele cu cea mai mare frecvenţă, respectiv cea mai mică frecvenţă de apariţie.

Exemplu. Pentru vectorul `x=('A','c','ă','a','s','t','a','e', 's','t','e','o','P','R','O','B','L','E','M','A','I','n', 't','e','r','e','s','a','n','t','a','d','e','L','U','C', 'R','U','C','u','v','e','c','t','o','r','s','I','a')`, litera care apar cel mai des în vector este `'E'` iar literele cele mai rar întâlnite sunt `'B', 'D', M', 'P', 'V'`

114. Se citeşte un vector cu n componente numere întregi. Să se determine cel mai mare număr raţional subunitar, în care numitorul şi numărătorul aparţin tabloului.

Exemplu. x=(2,1,3) => 2/3

115. Se citeşte de la tastatură un şir cu n elemente numere naturale, nu neapărat distincte. Să se afişeze elementul care apare de cele mai multe ori în şir. Dacă există mai multe astfel de elemente, se vor afişa toate.

Exemplu. Pentru n=8 şi elementele (23,7,11,7,19,7,11,11) se vor afişa elementele 7 şi 11 care apar fiecare de câte trei ori.

116. Se consideră n stâlpi de înălţimi h(1), h(2), ... h(n) metri. La baza fiecărui stâlp se află câte un melc codificat prin numărul stâlpului. Fiecare melc i urcă ziua p(i) metri şi coboară noaptea q(i) metri (p(i)>q(i)). Să se afişeze melcii în ordinea în care ating vârfurile stâlpilor. Se va considera că melcii care ajung în aceeaşi zi în vârf ajung simultan.

Exemplu. Pentru h=(10,5,8,3,25,7,9), p=(2,3,2,4,5,3,2), q=(1,2,1,3,1,2,0) melcii sosesc în următoarea ordine: 4, 2, 6, 7, 5, 3, 1.

117. Se dă un vector cu n componente numere întregi. Să se determine diferenţa maximă dintre două elemente consecutive ale acestui vector.

Exemplu. Dacă x=(15,36,94,15,64,32,245,11,260,33,56) diferenţa maximă este 249.

118. Se citesc vectorii A cu n elemente şi B cu m elemente. Să se determine care elemente din A sunt mai mici decât toate elementele din B.

Exemplu: Dacă a=(4,8,1,9,5,11,3,43,6,20) şi b=(9,9,6,9,9,8,6,9), atunci numărul căutat este 4, deoarece valorile 4,1,5 şi 3 sunt mai mici decât toate elementele lui b.

119. Să se determine vectorul y cu componentele $y_i, i = 1,2,\ldots,n$, unde y_i este poziţia valorii minime din şirul $x_i, x_{i+1}, \ldots, x_n$.

Exemplu. Pentru vectorul x=(9,4,3,8,5,1,6,2,7,10) se obţine vectorul y=(6,6,6,6,6,6,8,8,9,10).

120. Se citeşte de la tastatură un vector de n numere întregi şi o poziţie p. Toate elementele de la dreapta poziţiei p se adună la elementele aflate la stânga poziţiei p la aceeaşi distanţă faţă de poziţia p, ele devenind nule. Un element aflat la dreapta poziţiei p şi care nu are corespondent la stânga lui p rămâne nemodificat. Să se determine poziţia/poziţiile elementului/elementelor maxime din vectorul obţinut.

Exemplu. Pentru x=(15,36,94,15,64,32,245,11,260,33,56) şi p=4 rezultă x=(260,68,158,15,0,0,0,11,260,33,56) iar poziţiile elementului maxim în acest vector sunt 1 şi 9.

121. Să se determine cel mai mic număr prim aflat pe poziţii pare în vector. Dacă nu există un astfel de număr se va afişa un mesaj corespunzător.

Exemplu. Dacă `x=(9,31,38,5,62,44,38,17,19,38,50,74)` obţinem minimul `5`. Pentru vectorul `x=(9,4,62,81,31,74,697,26,12,351)` nu există un astfel de număr.

122. Fie un vector `x` de numere întregi. Să se determine un număr `p`, care să fie cel mai mare număr prim din cadrul vectorului. Dacă nu există numere prime în vectorul `x` atunci `p` va fi egal cu `0`. Dacă `p` nu este `0`, atunci să se împartă (ca numere întregi) toate componentele lui `x` la suma cifrelor lui `p`.

Exemplu. Dacă iniţial `x=(9,4,62,81,31,74,697,26,12,351)` atunci `p=31` iar vectorul va deveni `x=(2,1,15,20,7,18,174,6,3,87)`.

123. Se consideră un şir de `n` numere reale. Să se scrie programul care determină elementul maxim din subşirul format din termenii de rang impar ai şirului iniţial şi elementul minim din subşirul format din termenii de rang par ai şirului iniţial precum şi rangurile pe care apar aceste elemente în şir.

Exemplu. Dacă vectorul este `(2,84,9,1,2,6,48,312,4,78)` atunci minimul cerut este `1`, iar maximul este `48`.

124. Se consideră un şir de `n` numere întregi, nu toate egale şi fie `m` un număr natural dat, `m≥2`. Se împarte intervalul `(min,max)` în `m` subintervale de lungime egală, unde `min` şi `max` reprezintă valoarea minimă şi respectiv maximă din şir. Să se determine câte din elementele şirului se află în fiecare interval (intervalele se consideră deschise).

Exemplu. Pentru vectorul `(5,9,4,3,4.1,6,7,8,9,5,5.4)` şi `m=5` se obţin intervalele `(3,4.2)`, `(4.2,5.4)`, `(5.4,6.6)`, `(6.6,7.8)`, `(7.8,9)` în care se găsesc `2`, `2`, `1`, `1`, şi respectiv `1` elemente.

125. Să se formeze vectorul `z` cu componentele:

$$z_i = \begin{cases} \max\left\{\dfrac{x_1 + x_2 + \ldots + x_i}{i}, y_i\right\} & \text{daca } y_i \geq 0 \\ \max\left(x_i, \dfrac{y_1 + y_2 + \ldots + y_i}{i}\right) & \text{daca } y_i < 0 \end{cases}$$

pentru $i=1,2,\ldots,n$, unde $n \in N$, $x_i \in R$ şi $y_i \in R$ se citesc de la tastatură.

Exemplu. `x=(-5,6,-7,-9,9,41,25,31,-7,-41)`, `y=(98,74,-1,20,0,4,-33,-7,5,-9)` implică `z=(98,74,57,20,0,5.833,25,31,9.3333,15.1)`.

126. Să se scrie un program care calculează maximul dintre elementele pare ale unui vector de numere întregi.

Exemplu. Elementul maxim par din vectorul `(22,55,94,19,32,11,14,66,70,33)` este `94`.

127. Dându-se şirul x cu n componente reale, să se determine şi să se afişeze şirul y definit prin:

$$y_i = \begin{cases} \max\left(x_j \middle| x_j > 0, j = \overline{1,i}\right) & \text{daca i este par} \\ \dfrac{(x_1 + x_2 + \ldots + x_i)}{i} & \text{daca i este impar} \end{cases}, i=1,2,\ldots n.$$

Exemplu. Dacă x=(10,6,37,73,33,87,70,93,63,91) se obţine vectorul y=(10,10,17.6666,73,31.8,87,45.1428,93,52.4444,93).

128. Se dă n ∈ N şi vectorii $x = (x_1, x_2, \ldots, x_n)$ şi $y = (y_1, y_2, \ldots, y_n)$. Să se calculeze vectorul c astfel:

$$c_i = \begin{cases} \dfrac{(x_1 + x_2 + \ldots + x_i)}{i} & \text{daca } x_i y_i < 0 \\ \max(x_i, y_1, y_2, \ldots, y_i) & \text{daca } x_i y_i = 0, i=1,2,\ldots,n. \\ \min\{x_i, x_{i+1}, \ldots, x_n, y_i\} & \text{daca } x_i y_i > 0 \end{cases}$$

Exemplu. Dacă x=(5,-2,3,6,-4,8,-2,10,-3) şi y=(2,-3,-5,7,9,0,-4,0,-10) atunci c=(-4,-4,2,-4,1.6,9,-4,10,-10).

129. Dându-se n ∈ N şi $x_i, y_i \in R$ pentru $i = \overline{1, n}$ să se calculeze:

$$E = |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_n - y_n|$$

$$\text{şi } c_i = \begin{cases} \dfrac{x_i}{y_i * y_i + 1} & \text{daca } |x_i - y_i| < E \\ \max\{x_i, y_1, y_2, \ldots, y_i\} & \text{daca } |x_i - y_i| = E. \\ \min\{x_i, x_{i+1}, \ldots, x_n, y_i\} & \text{daca } |x_i - y_i| > E \end{cases}$$

Exemplu. Dacă x=(5,-2,3,6,-4,8,-2,10,-3) şi y=(2,-3,-5,7,9,0,-4,0,-10) atunci c=(1,-0.2,0,1154,0.12,-0.0488,8,-0.1176,10,-0.0297).

130. Se dau n ∈ N şi vectorii $x = (x_1, x_2, \ldots, x_n)$ şi $y = (y_1, y_2, \ldots, y_n)$, $x_i, y_i \in R$, $i = \overline{1, n}$ şi se cere să se calculeze şi să se afişeze pe ecran componentele vectorului c unde:

$$c_i = \begin{cases} \max(x_1, x_2, \ldots, x_i) & \text{daca } x_i < y_i \\ \max\{\min(x_1, x_2, \ldots, x_i), 0\} & \text{daca } x_i = y_i, i=1,2,\ldots,n. \\ \min(y_i, y_{i+1}, \ldots, y_n) & \text{daca } x_i > y_i \end{cases}$$

Exemplu. Dacă x=(5,2,3,7,4,-2,3,9,12,4). y=(10,4,3,2,4,5,3,-6,2,3) atunci c=(5,5,2,-6,2,7,0,-6,2,3).

131. Se consideră un şir cu n numere naturale. Se cere să se afişeze un număr din şir care are în scrierea sa cât mai multe cifre distincte. Dacă există mai multe numere cu această proprietate se va afişa primul dintre ele.

Exemplu. Pentru x=(54,12221,123,33,94) numărul cerut este 123.

132. Se dau $a \in R, n \in N$ şi perechile (x_i, y_i), i=1,2,...,n. de numere reale. Să se calculeze numărul m al valorilor y_i mai mari decât a şi

$$z_i = \begin{cases} \dfrac{x_1 + x_2 + \ldots + x_i}{i} & \text{daca } i < m \\ \max\{0, x_i, y_i\} & \text{daca } i = m. \\ |x_1 - y_1| + |x_2 - y_2| + \ldots + |x_i - y_i| & \text{daca } i > m \end{cases}$$

Exemplu. Dacă x=(2,8,3,6,10,5,8,3,4,1) şi y=(5,2,13,8,6,-5, 14,3,7,1) iar a=5 obţinem m=5 şi z=(2,5,4.3333,4.75,10,35, 41,41,44,44).

133. Se consideră un şir cu n numere naturale. Se cere să se afişeze un număr din şir pentru care restul împărţirii sale la un număr m dat de la tastatură este minim.

Exemplu. Pentru vectorul x=(12221,54,123,33,94) şi m=25 numărul cerut este 54.

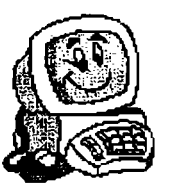
134. Să se calculeze:

$$z_i = \begin{cases} \dfrac{(x_1 + x_2 + \ldots + x_i)}{i} & \text{daca } x_i < y_i \\ 0 & \text{daca } x_i = y_i \\ \max(|y_i|, |y_{i+1}|, \ldots, |y_n|) & \text{daca } x_i > y_i \end{cases}$$

pentru i=1,2,...,n, unde $n \in N$ şi $x_i \in R, y_i \in R$ se citesc de la tastatură.

Exemplu. Dacă x=(5,2,3,7,4,-2,4,9,12,4) iar y=(10,4,3,-2,-4, 5,4,-6,3,2) rezultă z=(5,3.5,0,6,6,3.1666,0,6,3,2).

3.1.3. Căutări de elemente



135. Câte din elementele vectorului $x=(x_1, x_2, \ldots, x_n)$ de numere întregi sunt numere Fibonacci.

Exemplu. În vectorul x=(5,10,1,7,9,8,1,6,55,19) sunt 5 numere Fibonacci.

136. Scrieți un program care verifică dacă o valoare `x` se găsește sau nu printre cele `n` elemente ale unui șir dat, afișând un mesaj corespunzător. Valorile lui `n` și `x`, precum și elementele șirului se citesc de la tastatură. Se vor considera următoarele situații:

a) Se consideră șirul nesortat;

b) Șirul se consideră sortat iar căutarea se va face folosind algoritmul de căutare binară.

Exemplu. a) Dacă șirul este `(5,10,1,7,9,8,1,6,55,19)` atunci `x=33` nu se găsește în șir.

b) Dacă șirul este `(2,3,6,9,18,23,55,87,102)` atunci `x=87` se găsește pe poziția `8`.

137. Să se scrie un program care verifică dacă un număr `a` citit de la tastatură se găsește printre elementele de pe pozițiile pare ale unui vector `x`. Se vor considera următoarele cazuri:

a) se va face o căutare binară, presupunând că vectorul este sortat crescător iar numerele din vector sunt distincte două câte două;

Exemplu. În vectorul `(2,3,4,5,6,7)` numărul `6` nu se găsește pe pozițiile pare.

b) vectorul nu este sortat crescător, deci se va face o căutare secvențială.

Exemplu. În vectorul `(54,12,75,3,45,44,1)` numărul `12` se găsește pe poziția `2` iar `75` sau `22` nu se găsesc pe pozițiile pare.

138. Să se verifice dacă un vector reprezintă o mulțime (în sensul cunoscut de la matematică, adică numerele sunt distincte două câte două), dacă nu să se transforme acest vector în mod corespunzător.

Exemplu. Vectorul `x=(1,2,9,4,2,6,5,1,20,3)` nu este mulțime. Prin transformarea acestuia în mulțime se găsește `x=(1,2,9,4,6,5,20,3)`.

139. Se dau două șiruri de numere întregi, cu câte `n` elemente fiecare. Scrieți un program care testează dacă elementul maxim al primului șir se găsește în cel de-al doilea șir, tipărind `TRUE` sau `FALSE`. Se vor considera următoarele situații:

a) Se consideră vectorul al doilea nesortat;

Exemplu. Dacă `a=(1,9,4,3,7)` iar `b=(666,26,6,1,12,235,9,78,94,101)`, maximul din `a(9)` se găsește în `b` pe poziția `7`.

b) Vectorul al doilea se consideră sortat iar căutarea se va face folosind algoritmul de căutare binară.

Exemplu. Dacă `a=(1,9,4,3,7)` iar `b=(1,6,9,12,26,78,94,101,235,666)` maximul din `a(9)` se găsește în `b` pe poziția `3`.

140. O funcție $f: \{1,2,\ldots,m\} \to \{1,2,\ldots,n\}$ se poate reprezenta în calculator printr-un vector $F = (F_1, F_2, \ldots, F_m)$ cu m componente unde $F_i = f(i)$. Să se verifice dacă funcția f este surjectivă.

Exemplu. Funcția $f:\{1,2,3,4,5,6,7,8,9,10\} \to \{1,2,3,4\}$ definită prin vectorul $F=(1,2,4,2,1,1,1,4,2,2)$ nu este surjectivă pentru că nu există nici un $i \in \{1,2,3,4,5,6,7,8,9,10\}$ astfel încât $f(i)=3 \in \{1,2,3,4\}$.

141. Se dau $m, n \in N$ și mulțimile $A = \{a_1, a_2, \ldots, a_m\}$ și $B=\{b_1, b_2, \ldots, b_n\}$ memorate ca vectori. Să se scrie câte un program care calculează:

a) $C = A \cap B$;

b) $C = A \cup B$

c) $C = A - B$

Vectorii A și B se consideră nesortați.

Exemplu. Dacă $A=(1,5,2,9,12)$ și $B=(5,3,4,6,33,99,12,64)$ atunci a) $C=(5,12)$; b) $C=(1,5,2,9,12,3,4,6,33,99,64)$; c) $C=(1,2,9)$.

3.1.4. Subșiruri de elemente

142. Se dă un vector de numere întregi pozitive de cel puțin două cifre fiecare. Să se determine cea mai lungă subsecvență de elemente consecutive prime, ale căror oglindire sunt tot numere prime.

Exemplu. Dacă vectorul este (51,48,11,971,44,19,181,751,347, 33,929,359) secvența cerută este 181, 751, 347.

143. Se dă un vector de numere întregi. Se cere să se afișeze subsecvența palindromică de lungime maximă. (elementele subsecvenței sunt elemente consecutive în vector).

Exemplu. Dacă vectorul dat este (1,12,31,12,1,5,13,27,4,4,27,13, 9,26,9,13,27,4) atunci secvența palindromică de lungima maximă este (4,27,13,9,26,9,13,27,4).

144. Se dă un șir de numere. Găsiți un subșir de lungime maximă care începe și se sfârșește cu același număr.

Exemplu. Pentru vectorul (57,55,24,9,50,74,3,98,4,86,48,75,88, 86,29,88,16,82) secvența cerută este (86,48,75,88,86).

145. Se dau două şiruri de numere întregi x cu nx elemente şi y cu ny elemente, unde nx>ny. Să se decidă dacă y este un subşir al lui x, adică dacă există un număr k astfel încât

$$x_k = y_1,\ x_{k+1} = y_2,\ \ldots\ x_{k+ny-1} = y_{ny}.$$

Exemplu. Dacă x=(2,7,3,9,10,4,5,8,6,1,9) atunci şirul y=(9,10,4) este subşir al lui x dar y=(4,5,6,9) nu este subşir al lui x.

146. Definim noţiunea de pantă ca fiind o secvenţă de elemente consecutive din vectorul A astfel încât a[i]=a[i-1]+1. Să se determine panta de lungime maximă.

Exemplu. În vectorul (1,2,3,8,6,7,10,11,12,13,3,1,4,5) panta de lungima maximă este (10,11,12,13).

147. Fie un tablou a array[0..n – 1] cu elemente de tip întreg cu proprietatea a[0] ≤ a[1] ≤ ... ≤ a[n – 1]. Un platou este o subsecvenţă maximală de elemente consecutive în tablou care conţin aceeaşi valoare. Lungimea unui platou este numărul de elemente ce compun platoul. Să se scrie un program care afişează lungimea p a celui mai lung platou.

Exemplu. Pentru a=(2,4,4,4,5,6,8,8,8,8,9,9,10,14,14,15,16,16,16,17) cel mai lung platou începe în poziţia 7 şi are lungimea 4.

148. Se consideră un şir de n numere naturale mai mici decât 32000. Se cere să se determine o succesiune cu număr maxim de elemente din şir, care are proprietatea că fiecare componentă a suceesiunii are acelaşi număr de cifre distincte în scrierea sa.

Exemplu. Pentru vectorul (55,8,37,949,26,2213,157,38,2) secvenţa cerută este (37,949,26).

149. Pentru un vector x de n numere întregi dat, să se afişeze o secvenţă (dacă există) de elemente consecutive din vector pentru care suma elementelor se divide cu n.

Exemplu. Pentru vectorul (89,54,4,63,96,17,98,6,55,60) cea mai lungă secvenţă care îndeplineşte condiţia cerută este (63,96,17,98,6).

150. Se dau doi vectori A cu m componente şi B cu n componente (n>m). Componentele ce formează vectorul A sunt distincte două câte două. Să se scrie un program care să verifice dacă toate componentele vectorului A luate în orice ordine se găsesc în vectorul B pe poziţii consecutive. Dacă testul este reuşit să se afişeze fiecare poziţie din vectorul B unde începe secvenţa de elemente din A.

Exemplu. Dacă A=(3,4,5,6) şi B=(2,5,6,4,3,9,3,12) atunci vectorul A se găseşte în B începând cu poziţia 2.

150. Se dau doi vectori `A` cu `m` componente şi `B` cu `n` componente (`n>m`). Componentele ce formează vectorul `A` sunt distincte două câte două. Să se scrie un program care să verifice dacă toate componentele vectorului `A` luate în orice ordine se găsesc în vectorul `B` pe poziţii consecutive. Dacă testul este reuşit să se afişeze fiecare poziţie din vectorul `B` unde începe secvenţa de elemente din `A`.

Exemplu. Dacă `A=(3,4,5,6)` şi `B=(2,5,6,4,3,9,3,12)` atunci vectorul `A` se găseşte în `B` începând cu poziţia 2.

3.1.5. Sortări şi interclasări

151. *Sortarea prin metoda bulelor.* Metoda constă în parcurgerea vectorului, interschimbând valorile adiacente din vector atunci când acestea nu se află în ordinea dorită. Atunci când în urma unei astfel de treceri prin vector nu s-a făcut nici o interschimbare înseamnă că vectorul este sortat. Să se scrie un program care sortează crescător un vector cu `n` componente numere întregi folosind această metodă.

152. *Sortarea prin amestecare.* Este o variantă "uşor" îmbunătăţită a metodei bulelor. Ea constă în efectuarea de treceri alternative în direcţii opuse. Se poate observa că la parcurgerea de la stânga la dreapta valorile mari (în cazul sortării crescătoare) migrează spre dreapta iar la parcurgerea de la dreapta spre stânga cele mici migrează spre stânga.

153. *Metoda bulelor cu santinelă.* Este asemănătoare cu metoda bulelor. Diferenţa constă că la fiecare pas se memorează poziţia în care a avut loc ultima interschimbare. La pasul următor trecerea în vector se face doar până la această poziţie (numită santinelă), restul elementelor fiind evident sortate.

154. *Sortarea prin selecţie directă.* La pasul `i` al algoritmului (`i=1,2,...,n`, unde `n` este lungimea vectorului) al acestei metode se determină

$$x_k=\min\{x_i, x_{i+1}, \ldots, x_n\}$$

şi se aduce această valoare pe poziţia x_i, prin interschimbarea valorilor x_i şi x_k.

155. *Sortarea prin numărare.* Constă în a număra pentru fiecare element x_i câte elemente mai mici decât el există în vector; numerele obţinute se memorează într-un vector `c`. Pe baza lui vom rearanja elementele lui `x` astfel: elementul `x[i]` trebuie pus pe poziţia `c[i]+1`.

Atenţie! Acest algoritm funcţionează corect doar pentru valori distincte în vector. El poate fi însă modificat pentru a funcţiona şi pentru cazul general. Cum?

156. Se citesc n numere întregi. Să se afişeze cel mai mare număr obţinut din prima cifră a fiecărui număr.

Exemplu. 900, 265, 75 => 972.

157. Cunoscând numele, înălţimea şi greutatea a n elevi (1≤n≤100) să se afişeze numele acestora în ordinea descrescătoare a înălţimii. Dacă există doi elevi cu aceeaşi înălţime se va afişa primul cel care are greutatea mai mare.

Exemplu. Dacă numele elevilor sunt ('Florin' 'Emilia', 'Eduard', 'Alina', 'Marius'), înălţimile lor sunt (130,133,130,145,123) iar greutăţile lor sunt (30,25,29,34,26) atunci elevii se vor afişa în ordinea **Alina, Emilia, Florin, Eduard, Marius.**

158. Construiţi un algoritm care determină şi afişează cel mai mare număr care se poate forma având exact aceleaşi cifre ca şi un număr natural n citit de la tastatură (n≤999.999.999). Scrieţi programul corespunzător.

Exemplu. Pentru n=306, cel mai mare număr natural având aceleaşi cifre ca şi n este 630.

159. Se consideră un vector conţinând doar elemente 0 şi 1. Să se ordoneze acest vector parcurgându-l o singură dată şi fără a folosi alţi vectori.

Exemplu. Pentru vectorul (1,1,0,0,0,1,0,0,1,1,1,0,1,0,0) se va obţine (0,0,0,0,0,0,0,0,1,1,1,1,1,1,1).

160. Se citesc de la tastatură n numere naturale. Se cere să se sorteze şirul în ordinea crescătoare a numărului de apariţii a cifrei 1 în reprezentarea în baza 2 a numerelor. În cazul în care două numere au acelaşi număr de cifre egale cu 1 în reprezentarea lor binară atunci ele vor apărea în ordinea în care sunt citite de la tastatură.

Exemplu. Vectorul (5,32,71,1,14,13,67,82,35) va fi după sortare (32,1,5,14,13,67,82,35,71).

161. Fiind dat vectorul x cu n componente numere întregi să se ordoneze crescător primele k numere şi descrescător celelalte numere, pentru k dat, $k \in \{1,2,3,\ldots,n\}$.

Exemplu. Pentru x=(45,95,12,34,2,154,26,24,132,14,453) şi k=4 se obţine x=(12,34,45,95,453,154,132,26,24,14,2).

162. Se consideră tipul Culoare definit prin:

```
type Culoare=(rosu, galben, albastru);
```

şi un tablou u:array[1..n] of Culoare. Se presupune n≥1. Se cere să se scrie un program care afişează tabloul u cu valorile rearanjate prin interschimări astfel încât elementele egale cu galben să se găsească după elementele egale cu roşu şi înaintea celor egale cu albastru.

Exemplu. Dacă vectorul este `x=(albastru,rosu,rosu,galben,rosu, albastru)` se obține `x=(rosu,rosu,rosu,galben,albastru, albastru)`.

163. Se dau două mulțimi prin intermediul a doi vectori, `x` și `y`, cu `n` (`1≤n≤100`) componente întregi. Se cere să se verifice dacă aceste mulțimi sunt proporționale. În caz afirmativ se va afișa și șirul de rapoarte egale.

Exemplu.

Pentru n=3 și mulțimile	Se va afișa
20 2 4	Mulțimi proporționale
3 6 30	2/3=4/6=20/30

164. Se dă un șir de numere întregi. Să se ordoneze șirul astfel încât primele poziții să conțină componentele pare ordonate crescător, iar apoi să conțină componentele impare ordonate descrescător.

Exemplu. Dacă vectorul inițial este `(45,30,3,154,23,2,28,14,453)` după sortare se obține `(2,14,28,30,154,453,45,23,3)`.

165. Dându-se un vector cu maxim `100` de numere naturale, să se localizeze elementul maxim și toate elementele dinaintea lui să se ordoneze crescător, iar cele de după el, descrescător.

Exemplu. Dacă vectorul inițial este `(3,1,9,4,100,2,36,1,4,9,5)` după sortare se va obține `(1,3,4,9,100,36,9,5,4,2,1)`.

166. Se consideră un vector conținând `n` (`n<=100`) numere naturale cu valori cuprinse între `1` și `60000`. Să se alcătuiască un program care să ordoneze crescător doar numerele pare din vector, fără însă a afecta pozițiile pe care sunt situate numerele impare. Programul va afișa pe ecran vectorul obținut după ordonare.

Exemplu. `X=(11,40,32,44,3,13,8,17)` => `x=(11,8,32,40,3,13, 44,17)`.

167. Se consideră un vector cu `n` componente. Știind că el conține două subsecvențe de numere ordonate crescător, să se ordoneze întregul vector prin interclasarea celor două subsecvențe.

Exemplu. `X=(1,3,5,7,20,4,9,11,44,60)` => `X=(1,3,4,5,7,9,11, 20,44,60)`

168. Se dă un șir de numere, scrise într-o ordine aleatoare. Ele reprezintă rezultatele parțiale ale unor operații de adunare cu numere naturale. Printre numere se găsește și numărul de la care s-a plecat inițial. Să se reconstituie exercițiul folosind numerele date.

Exemplu. Dacă numerele date sunt: `12`, `6`, `23` atunci exercițiul este următorul: `6+6=12`; `12+11=23`.

169. Se dă un vector cu elemente având doar valorile 1, 2, 3. Vectorul se consideră ordonat "parțial": primele k-1 poziții sunt ocupate de elemente egale cu 1, pozițiile de la k la l-1 de elemente egale cu 2, pozițiile de la l la m-1 de elemente egale cu 3, iar pozițiile de la m la n de elemente neordonate (1<=k<=l<=m<=n, n,m,k,l dați). Să se scrie un program care sortează vectorul dat. Sortarea se va face prin interschimbarea valorilor între ele. Valoarea unui element se analizează o singură dată. În afara celor n locații pentru șirul dat, nu se va folosi nici o variabilă auxiliară pentru interschimbarea valorilor între ele.

Exemplu. Dacă inițial vectorul este: (1,1,1,1,2,2,2,2,2,2,3,3,2,3, 1,3,2,2,3), adică n=19, k=5, l=11, m=13, atunci în final ordinea acestor pioni va fi (1,1,1,1,1,2,2,2,2,2,2,2,2,2,3,3,3,3,3).

170. Se dau m, n ∈ N și mulțimile A = $\{a_1, a_2, \ldots, a_m\}$ și B=$\{b_1, b_2, \ldots, b_n\}$ memorate ca vectori. Să se scrie câte un program eare calculează:

a) C = A ∩ B

b) C = A ∪ B

c) C = A - B

Cei doi vectori se consideră sortați crescător.

Exemplu. A=(1,5,7,9,10), B=(5,6,9,12,14) se obține

a) C=(5,9);

b) C=(1,5,6,7,9,10,12,14);

c) C=(1,7,10).

171. Să se ordoneze crescător un vector conform următoarelor reguli:

se ordonează crescător prima treime a vectorului folosind inserția directă;

se ordonează crescător a doua treime a vectorului folosind metoda bulelor;

se ordonează descrescător ultima treime a vectorului folosind selecția directă;

se interclasează cele trei segmente sortate ale vectorului.

Se vor folosi maxim doi vectori.

Exemplu. Dacă inițial vectorul era (1,5,2,4,3,9,6,7,15,8,3,13) după cele trei sortări intermediare vectorul va fi (1,2,4,5,3,6,7,9,15, 13,8,3) iar după interclasarea celor trei secvențe (1,2,3,3,4,5,6,7,8, 9,13,15).

172. Se citește de la tastatură un șir de n numere naturale cuprinse între 1 si 60000. Se cere ordonarea crescătoare a acestui șir folosind următoarea

3.2. Tablouri bidimensionale (matrice)

3.2.1. Parcurgeri (sume numărări)

173. Se dau două matrice cu `m` linii şi `n` coloane. Să se verifice dacă ele sunt egale.

Exemplu. Matricele $\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$ şi $\begin{pmatrix} 1 & 3 & 5 \\ 2 & 0 & 6 \end{pmatrix}$ nu sunt egale.

174. Fiind dată o matrice a cu `m` linii şi `n` coloane şi elemente numere întregi, să se construiască transpusa acesteia (matricea care se obţine scriind liniile pe coloane şi invers).

Exemplu: Transpusa matricei `a` = $\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$, va fi $\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$.

175. Se dă un tablou bidimensional `a` cu `m` linii şi `n` coloane (`1≤m,n≤50`) componente numere întregi şi un număr întreg `k`. Se cere să se afişeze tabloul `a` cu componentele mărite cu `k`.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$ şi `k=3` se obţine matricea $\begin{pmatrix} 4 & 6 & 8 \\ 5 & 7 & 9 \end{pmatrix}$.

176. Să se scrie un program care înmulţeşte o matrice `A` cu `m` linii şi `n` coloane cu o constantă `k`.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$ şi `k=3` se obţine matricea $\begin{pmatrix} 3 & 9 & 15 \\ 6 & 12 & 18 \end{pmatrix}$.

177. Să se calculeze suma elementelor pozitive şi impare dintr-o matrice `A` cu `m` linii şi `n` coloane.

Exemplu. Dacă `A` = $\begin{pmatrix} 1 & 5 & 2 & 8 \\ -5 & 7 & 4 & -2 \\ 12 & 15 & 6 & 9 \end{pmatrix}$ atunci suma este 37.

178. Scrieţi un program care testează dacă o matrice `A` cu `n` linii şi `n` coloane este triunghiulară inferior sau nu, faţă de una din cele două diagonale ale matricei, adică toate elementele de deasupra diagonalei considerate sunt nule.

Exemplu. Matricea $\begin{pmatrix} 0 & 0 & 0 & 6 \\ 0 & 0 & 6 & 0 \\ 0 & 2 & 9 & 4 \\ 3 & 6 & 4 & 0 \end{pmatrix}$ este triunghiulară inferior faţă de diagonala secundară.

179. Se citeşte de la tastatură o matrice `A` cu `m` linii şi `n` coloane. Să se afişeze produsul elementelor pozitive aflate pe linii pare şi coloane impare.

Exemplu. Pentru $\begin{pmatrix} 5 & 2 & 1 & -3 & 7 & -2 \\ 1 & 5 & -2 & -11 & 3 & 0 \\ 3 & 31 & 6 & -4 & 18 & 9 \\ 8 & 8 & 5 & 2 & -1 & 6 \end{pmatrix}$ se obţine produsul 120 (=1·3·8·5)

180. Fiind dată o matrice a cu `m` linii şi `n` coloane cu elemente numere întregi, scrieţi un program care afişează numărul elementelor pozitive, numărul elementelor negative şi numărul elementelor nule din matrice.

Exemplu. În matricea $\begin{pmatrix} 1 & -2 & 3 & -4 & 0 & 1 \\ 5 & 2 & 0 & 10 & 9 & -4 \\ 0 & 0 & 1 & 0 & 12 & 3 \\ -5 & 1 & 35 & 8 & -21 & 0 \end{pmatrix}$ există 6 elemente nule, 13 elemente pozitive şi 5 elemente negative.

181. Pentru o matrice `A` cu `m` linii şi `n` coloane să se verifice dacă ea conţine cel puţin o linie simetrică, adică elementele egal depărtate de capetele liniei respective sunt egale.

Exemplu. În matricea $\begin{pmatrix} 1 & 3 & 0 & 3 & 1 \\ 5 & 9 & 1 & 9 & 5 \\ 4 & 9 & 7 & 3 & 2 \end{pmatrix}$ primele două linii sunt simetrice.

182. Fiind dată o matrice `m` linii şi `n` coloane cu elemente numere întregi, scrieţi un program care determină media aritmetică a elementelor matricei.

Exemplu. Pentru matricea $\begin{pmatrix} 2 & 9 & 4 & 3 \\ -2 & 1 & 5 & 7 \\ 6 & 10 & 8 & -12 \\ 0 & 31 & 5 & 11 \end{pmatrix}$ se obţine media 5.5.

183. Se să un tablou bidimensional cu `m` linii şi `n` coloane (`1≤m,n≤50`) având componentele întregi. Se cere să se calculeze suma componentelor de pe marginea tabloului.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 2 & 3 \\ 5 & 6 & 7 \\ 9 & 10 & 11 \\ 13 & 14 & 15 \end{pmatrix}$ se obţine suma 80.

184. Se citeşte o matrice `A` cu `m` linii şi `n` coloane, conţinând caracterele `'0'`, `'1'`, `'9'`. Să se afişeze numerele ce se obţin din cifrele fiecărei linii.

Exemplu. `A` = $\begin{pmatrix} '0' & '0' & '1' & '2' \\ '0' & '4' & '0' & '7' \end{pmatrix}$ se va afişa 12, 407.

185. Fiind dată o matrice `a` cu `m` linii şi `n` coloane cu elemente numere întregi, scrieţi un program care tipăreşte elementele pare aflate pe linii impare şi coloane divizibile cu 3.

Exemplu. Pentru matricea

$$\begin{pmatrix} 1 & 38 & 2 & 4 & 54 & 8 & 62 & 177 & 7 \\ 10 & 11 & 212 & 13 & 14 & 15 & 16 & 17 & 18 \\ 21 & 22 & 91 & 23 & 24 & 6 & 25 & 26 & 3 \\ 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 & 37 \end{pmatrix}$$

se vor afişa numerele 2, 8, 6.

186. Să se scrie un program care însumează elementele de deasupra diagonalei principale, inclusiv diagonala, dintr-o matrice pătratică.

Exemplu. Pentru matricea $\begin{pmatrix} 2 & 9 & 4 & 3 \\ -2 & 1 & 5 & 7 \\ 6 & 10 & 8 & -12 \\ 0 & 31 & 5 & 11 \end{pmatrix}$ suma este 38.

187. Se dă un tablou bidimensional cu `n` linii şi `n` coloane (`1≤n≤10`) de componente întregi. Se cere să se calculeze suma componentelor ce se află pe pătrate concentrice ale tabloului, dinspre exterior spre interior.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 6 & 7 & 8 \\ 1 & 3 & 4 & 6 \\ 1 & 0 & 5 & 4 \\ 7 & 3 & 2 & 7 \end{pmatrix}$ se obţin sumele 53 şi 12.

188. Să se scrie un program care însumează elementele de sub diagonala principală, exclusiv diagonala, dintr-o matrice pătratică.

Exemplu. Pentru matricea $\begin{pmatrix} 2 & 9 & 4 & 3 \\ -2 & 1 & 5 & 7 \\ 6 & 10 & 8 & -12 \\ 0 & 31 & 5 & 11 \end{pmatrix}$ suma este 50.

189. Să se scrie un program care înmulţeşte la dreapta o matrice dreptunghiulară `A`, cu un vector.

Exemplu. Dacă $\mathtt{A} = \begin{pmatrix} 1 & 2 & 3 & 5 & 1 \\ 3 & 2 & 11 & 7 & 3 \\ 4 & 6 & 8 & 9 & 12 \end{pmatrix}$ şi $\mathtt{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix}$ atunci $\mathtt{Ax} = \begin{pmatrix} 39 \\ 83 \\ 136 \end{pmatrix}$.

190. Să se scrie un program care însumează elementele de deasupra diagonalei secundare, exclusiv diagonala, dintr-o matrice pătratică.

Exemplu. Pentru matricea $\begin{pmatrix} 2 & 9 & 4 & 3 \\ -2 & 1 & 5 & 7 \\ 6 & 10 & 8 & -12 \\ 0 & 31 & 5 & 11 \end{pmatrix}$ suma este 20.

191. Să se construiască o matrice `A` cu `n` linii şi `n` coloane ce se completează cu termenii şirului lui Fibonacci. Completarea se va face în ordinea liniarizării pe linii a matricei. Nu se vor folosi structuri de date auxiliare.

Exemplu. Pentru `n=3` se va afişa matricea $\begin{pmatrix} 1 & 1 & 2 \\ 3 & 5 & 8 \\ 13 & 21 & 34 \end{pmatrix}$.

192. Se dă un tablou binimensional cu `m` linii şi `n` coloane (`1≤m,n≤50`) având componente binare. Fiecare linie constituie reprezentarea câte unui număr natural în baza 2. Se cere să se afişeze aceste numere.

Exemplu. Dacă matricea este $\begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}$ se vor afişa valorile `77`, `55`, `117`.

193. Să se elimine dintr-o matrice `A` (cu `m` linii şi `n` coloane) linia `l` şi coloana `k` şi să se listeze matricea rămasă. (Nu se va folosi o altă matrice).

Exemplu. Dacă matricea iniţială este $\begin{pmatrix} 2 & 9 & 4 & 3 \\ -2 & 1 & 5 & 7 \\ 6 & 10 & 8 & -12 \\ 0 & 31 & 5 & 11 \end{pmatrix}$ iar `l=1`, `k=3` atunci matricea finală va fi $\begin{pmatrix} -2 & 1 & 7 \\ 6 & 10 & -12 \\ 0 & 31 & 11 \end{pmatrix}$.

194. Fiind dată matricea `A` cu `m` linii şi `n` coloane să se elimine liniile şi coloanele la a căror intersecţie se găseşte valoarea `v` dată de la tastatură.

Exemplu. Dacă matricea iniţială este $\begin{pmatrix} 1 & 7 & 3 & 8 & 12 \\ 2 & 15 & 7 & 9 & 4 \\ 5 & 9 & 35 & 2 & 3 \\ 31 & 6 & 14 & 0 & 25 \end{pmatrix}$ iar valoarea de eliminat este `3` se obţine matricea $\begin{pmatrix} 2 & 15 & 9 \\ 31 & 6 & 0 \end{pmatrix}$.

195. Să se insereze o linie `l` în poziția p_1 și o coloană `c` în poziția p_2, elementele acestora fiind citite de la tastatură. Nu se vor folosi structuri de date auxiliare.

Exemplu. Dacă matricea inițială este $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ și `p1=2, p2=1, l=(11,11, 11), c=(10,10,10,10)` atunci matricea va fi în final $\begin{pmatrix} 10 & 1 & 2 & 3 \\ 10 & 11 & 11 & 11 \\ 10 & 4 & 5 & 6 \\ 10 & 7 & 8 & 9 \end{pmatrix}$.

196. Se dă o matrice cu `n` linii și `n` coloane cu componente întregi. Să se verifice dacă tabloul este simetric față de diagonala principală.

Exemplu. Matricea $\begin{pmatrix} 1 & 5 & 26 \\ 5 & 14 & 9 \\ 26 & 9 & 10 \end{pmatrix}$ este simetrică față de diagonala principală.

197. Se dă o matrice cu `n` linii și `n` coloane cu componente întregi. Să se verifice dacă tabloul este simetric față de diagonala secundară.

Exemplu. Matricea $\begin{pmatrix} 45 & 8 & 1 \\ 36 & 9 & 8 \\ 14 & 36 & 45 \end{pmatrix}$ este simetrică față de diagonala secundară.

198. Se dau numerele naturale `M` și `N` și un șir de numere reale `x[i]`, `i=1,2,...,M × N`. Să se genereze matricea `A`, de dimensiuni `M` și `N`, definită prin $a(i, j) = \dfrac{\sum_{k=1}^{i \times j} x_k}{(i \quad j - j + 1)}$ pentru $i = \overline{1, M}$, $j = \overline{1, N}$.

Exemplu. Pentru `m=4, n=3` și `x=(1,2,3,4,5,6,7,8,9,10,11,12)` se obține matricea $A = \begin{pmatrix} 1 & 3 & 6 \\ 1.5 & 3.33 & 5.25 \\ 2 & 4.2 & 6.4286 \\ 2.5 & 5.1429 & 7.8 \end{pmatrix}$

199. Se consideră un tabel `T` de dimensiune `m×n` și numărul natural `k` (`1≤k≤n`). Alcătuiți un program care transferă primele `k` coloane la sfârșitul tabelului, fără a utiliza alte structuri de date.

Exemplu. $T = \begin{pmatrix} 1 & 3 & 5 & 7 & 9 & 2 & 4 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{pmatrix}$, `k=3` => $T = \begin{pmatrix} 7 & 9 & 2 & 4 & 1 & 3 & 5 \\ 4 & 5 & 6 & 7 & 1 & 2 & 3 \\ 5 & 6 & 7 & 8 & 2 & 3 & 4 \end{pmatrix}$

200. Se dă o matrice pătratică de ordinul `N`. Să se formeze un vector cu `N` componente, în care componenta `i` a vectorului să fie egală cu raportul dintre suma elementelor din linia `i` şi suma elementelor din coloana `i`. Dacă suma elementelor de pe coloana `i` este nulă atunci componenta a `i`-a din vector va fi `0`.

Exemplu. Dacă matricea este $\begin{pmatrix} 5 & 0 & 6 & 1 \\ 2 & -5 & 8 & 3 \\ 2 & 0 & 4 & 0 \\ -1 & 5 & 2 & 7 \end{pmatrix}$ se va obţine vectorul `x=(1.5,0,0.3,1.1818)`.

201. Se dă o matrice `A` cu `m` linii şi `n` coloane. Să se afişeze matricea obţinută din `A` prin eliminarea liniilor şi coloanelor la intersecţia cărora se află un număr natural al cărui scriere în baza 2 are cel puţin două cifre egale cu `1`.

Exemplu. Dacă iniţial $A = \begin{pmatrix} 2 & 0 & 8 & 16 & 8 \\ 32 & 5 & 8 & 2 & 4 \\ 64 & 1 & 0 & 9 & 0 \\ 4 & 2 & 16 & 1 & 8 \end{pmatrix}$ atunci după eliminări (liniile 2 şi 3 şi coloanele 2 şi 4) se obţine matricea $A = \begin{pmatrix} 2 & 8 & 8 \\ 4 & 16 & 8 \end{pmatrix}$.

202. Pornind de la matricea $A = (a_{ij})_{\substack{i=\overline{1,m} \\ j=\overline{1,n}}}$ să se formeze matricea `B` cu `m` linii şi `n` coloane unde:

$$b_{ij} = \begin{cases} a_{ij} & \text{daca } i < j \\ \dfrac{a_{i1} + a_{in} + a_{1j} + a_{mj}}{4} & \text{altfel} \end{cases}$$

Exemplu. Pentru `m=3, n=5` şi $A = \begin{pmatrix} 1 & 5 & 3 & 2 & 9 \\ 7 & 6 & 4 & 8 & 1 \\ 12 & 9 & 40 & 2 & 321 \end{pmatrix}$ se obţine matricea $B = \begin{pmatrix} 5.75 & 5 & 3 & 2 & 9 \\ 5.25 & 6 & 4 & 8 & 1 \\ 86.5 & 86.75 & 40 & 2 & 321 \end{pmatrix}$.

203. Se citeşte de la tastatură o matrice cu `m` linii şi `n` coloane cu elemente numere întregi. Să se memoreze într-un vector `b` sumele elementelor de pe fiecare linie a matricii (`b[i]` va reprezenta suma elementelor de pe linia `i` în matrice).

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ se va obţine vectorul `b=(6,15,24)`.

204. Pentru numărul impar n, citit de la tastatură, să se construiască matricea pătrată de ordinul n, A=($a_{i,j}$) cu elemente întregi, astfel:

$$a_{ij} = \frac{i(n-1)+(n+1)j}{2} + \begin{cases} \frac{1-n}{2} & \text{daca } i+j=\text{par} \\ \frac{n-1}{2} & \text{daca } i+j=\text{impar} < n+1 \text{ si } i<j \\ \frac{3-n}{2} & \text{daca } i+j=\text{impar} \geq n+1 \text{ si } i>j \\ \frac{n+1}{2} & \text{daca } i+j=\text{impar} < n+1 \text{ si } i \geq j \\ \frac{1-n-2n}{2} & \text{daca } i+j=\text{impar} \geq n+1 \text{ si } i \leq j \end{cases}$$

Exemplu. Pentru n=3 se obţine matricea $\begin{pmatrix} 2 & 6 & 6 \\ 6 & 5 & 4 \\ 4 & 7 & 8 \end{pmatrix}$.

205. Să se verifice dacă o matrice pătratică A de ordinul n este matrice distanţă, adică:

$$\begin{cases} a_{ij} > 0 & \forall i,j \in \overline{1,n},\ i \neq j \\ a_{ii} = 0 & \forall i \in \overline{1,n} \\ a_{ij} = a_{ji} & \forall i,j \in \overline{1,n} \\ a_{ij} + a_{jk} \geq a_{ik} & \forall i,j,k \in \overline{1,n} \end{cases}$$

Exemplu. Matricea $\begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 3 \\ 2 & 3 & 0 \end{pmatrix}$ este o matrice distanţă.

206. Să se bordeze o matrice A (având m linii şi n coloane) cu linia m+1 şi coloana n+1, unde A[m+1,j] să fie suma elementelor de pe coloana j, iar A[i,n+1] să fie suma elementelor de pe linia i.

Exemplu. Pentru matricea iniţială $\begin{pmatrix} 1 & 5 & 3 \\ 4 & 7 & 2 \end{pmatrix}$ se obţine matricea

$$\begin{pmatrix} 1 & 5 & 3 & 9 \\ 4 & 7 & 2 & 13 \\ 5 & 12 & 5 & 0 \end{pmatrix}.$$

207. Să se verifice dacă o matrice pătrată `A` de ordinul `n` este pătrat magic sau nu. O matrice pătrată este pătrat magic dacă sumele de pe fiecare linie, coloană şi de pe cele două diagonale sunt egale.

Exemplu. Matricea $\begin{pmatrix} 4 & 3 & 8 \\ 9 & 5 & 1 \\ 2 & 7 & 6 \end{pmatrix}$ este un pătrat magic.

208. O metodă foarte simplă de generare a unui pătrat magic constă în completarea pe rând cu valorile `1,2,...,n`^{2} astfel:

se plasează `1` în centrul ultimei coloane;

se merge în linie oblică, dreapta-jos, cu numărul următor; dacă se iese din pătrat prin partea dreaptă, se merge în partea opusă, în stânga liniei unde trebuia depus numărul, iar dacă se iese prin partea de jos a pătratului, se merge în partea de sus a coloanei unde trebuia depus numărul;

după ce se completează un grup de `n` numere, se merge cu o căsuţă spre stânga, pe aceeaşi linie, pentru a se repeta apoi pasul `2` şi a genera următorul grup de `n` numere.

Să se scrie un program care implementează acest algoritm.

Exemplu. Pentru `n=5` se obţine în final următorul pătrat magic:

$$\begin{pmatrix} 11 & 10 & 4 & 23 & 17 \\ 18 & 12 & 6 & 5 & 24 \\ 25 & 19 & 13 & 7 & 1 \\ 2 & 21 & 20 & 14 & 8 \\ 9 & 3 & 22 & 16 & 15 \end{pmatrix}.$$

209. Pentru o matrice `A` cu `m` linii şi `n` coloane să se afişeze coordonatele tuturor vârfurilor. Un vârf este un element al matricei care are toţi vecinii cu mai mici decât el.

Exemplu. În matricea $\begin{pmatrix} 1 & 97 & 1 & 6 \\ 6 & 7 & 3 & 40 \\ 2 & 1 & 6 & 15 \\ 0 & 2 & 83 & 1 \end{pmatrix}$ vârfurile sunt `a[1,2]=97`, `a[2,4]=40` şi `a[4,3]=83`.

210. Fie `A` o matrice cu `m` linii şi `n` coloane, cu numere reale şi un vector `v` cu `n` componente reale. Să se determine dacă acest vector apare ca linie în matricea `A` şi, în caz afirmativ, să se afişeze numărul acestei linii.

Exemplu. Dacă `A` = $\begin{pmatrix} 1 & 3 & 6 & 2 & 9 \\ 7 & 4 & 0 & 3 & 8 \\ 5 & 2 & 1 & 47 & 6 \end{pmatrix}$ iar `v=(5,2,1,47,6)` atunci se va afişa valoarea `3`.

211. Se citeşte de la tastatură o matrice a cu n linii şi n coloane cu elemente numere întregi. Să se copieze într-un vector elementele pozitive aflate deasupra diagonalei principale în matrice, apoi să se tipărească vectorul.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 5 & -3 \\ -4 & 7 & 2 \\ 14 & 33 & 9 \end{pmatrix}$ se obţine vectorul (5,2).

212. Se dă o matrice A, pătratică de ordinul n. Să se genereze vectorul x având n componente definite astfel:

$$x[i] = \begin{cases} 1 & \text{daca } \sum_{j=1}^{i} a_{ij} > \sum_{j=1}^{i} a_{ji} \\ 0 & \text{in caz contrar} \end{cases}$$

Exemplu. Pentru $A = \begin{pmatrix} 1 & 2 & 9 & 2 & 2 \\ 5 & 6 & 7 & 3 & 6 \\ 4 & 3 & 6 & 1 & 1 \\ 1 & 5 & 2 & 3 & 9 \\ 7 & 0 & 10 & 3 & 4 \end{pmatrix}$ se obţine vectorul x=(0,1,0, 1,1).

213. Fie A o matrice pătratică de ordinul n. Să se scrie un program care generează o matrice B (pătratică de ordinul n) ale cărei elemente sunt definite prin relaţia $b_{ij}=(a_{ij}+a_{ji})/2$ efectuând un număr minim de paşi.

Exemplu. Pentru $A = \begin{pmatrix} 0 & 0 & 14 & 6 \\ 2 & 2 & 0 & 4 \\ 0 & 8 & 3 & 6 \\ 3 & 8 & 7 & 4 \end{pmatrix}$ se va obţine

$$B = \begin{pmatrix} 0 & 1 & 7 & 4.5 \\ 1 & 2 & 4 & 6 \\ 7 & 4 & 3 & 6.5 \\ 4.5 & 6 & 6.5 & 4 \end{pmatrix}.$$

214. Se dă un tablou bidimensional a cu m linii şi n coloane ($1 \le m,n \le 50$) de componente întregi. Să se afişeze numărul de ordine al liniilor cu componente aflate în ordine strict crescătoare din a.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 2 & 6 & 9 & 33 \\ 2 & 4 & 3 & 1 & 9 \\ 1 & 6 & 11 & 11 & 21 \\ 6 & 9 & 11 & 12 & 22 \end{pmatrix}$ se vor afişa numerele 1, 4.

215. Se dau vectorii B cu m elemente şi D cu n elemente reale. Se cere să se determine matricea $A = (a_{ij})_{\substack{i=1,m \\ j=1,n}}$, obţinută după următorul algoritm:

a) i=1, j=1

b) a_{ij}=min{bi,d_j}

b_i=b_i-a_{ij}

d_j=d_j-a_{ij}

c) dacă b_i=0 atunci i=i+1

altfel j=j+1

d) daca i≤m şi j≤n se continuă cu pasul b.

Exemplu. Pentru m=5, n=8, B=(5,9,2,1,3), D=(3,2,5,7,10,4,-2,1) se obţine matricea

$$A = \begin{pmatrix} 3 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5 & 4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0 \end{pmatrix}$$

216. Se dă o matrice A cu m linii şi n coloane cu elemente 0 şi 1. Să se scrie un program care să se afişeze toate perechile de linii complementare din matrice. Două linii se numesc complementare dacă suma componentelor de pe aceeaşi coloană este 1.

Exemplu. În matricea $\begin{pmatrix} 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \end{pmatrix}$ liniile complementare sunt (1,3) şi respectiv (2,5).

217. Se dă un tablou bidimensional a cu m linii şi n coloane (1≤m,n≤50) de componente întregi. Să se permute circular cu k (1≤k≤50) poziţii în jos coloanele lui a.

Exemplu. Pentru k=2, şi matricea iniţială $A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ se va obţine după permutare matricea $A = \begin{pmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \\ 1 & 2 & 3 \end{pmatrix}$.

218. Scrieți un program care verifică dacă într-o matrice `A` cu `m` linii și `n` coloane liniile `L1` și L2 sunt identice sau nu.

Exemplu. În matricea $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 5 & 6 & 7 \\ 1 & 2 & 3 & 4 \\ 8 & 3 & 1 & 56 \end{pmatrix}$ liniile `1` și `3` sunt identice.

219. Să se completeze în spirală o matrice cu `n` linii și `n` coloane cu primele n^2 numere întregi cu cel puțin 2 cifre, având cu proprietatea că suma cifrelor lor este divizibilă cu `5`.

Exemplu. Pentru `n=4` se obține matricea $\begin{pmatrix} 14 & 19 & 23 & 28 \\ 69 & 73 & 78 & 32 \\ 64 & 87 & 82 & 37 \\ 55 & 50 & 46 & 41 \end{pmatrix}$.

220. Pentru o matrice cu `m` linii și `n` coloane, cu elemente numere reale, să se listeze toate liniile, precum și numerele lor de ordine, care conțin cel puțin `k` elemente nule.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 2 & 0 & 0 & 1 & 6 \\ 0 & 2 & 9 & 0 & 0 & 1 \\ 9 & 5 & 0 & 1 & 24 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix}$, `k=2` se afișează liniile `1,2`.

221. Se consideră tabelul `T` cu `n` linii și coloane, cu elemente `0` și `1`. Mai mult, toate elementele de pe diagonala principală sunt egale cu `0`. Alcătuiți un program în cadrul căruia se determină toate perechile `(i,j)`, unde `i` este numărul liniei iar `j` numărul coloanei cu proprietatea că toate elementele din linia `i` sunt `0` iar cele din coloana `j` (cu excepția elementului de pe linia `i`) sunt toate `1`. Pot exista mai multe astfel de soluții?

Exemplu. În matricea $\begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}$ singura pereche este `(2,3)`.

222. Se dă un tablou binimensional cu `m` linii și `n` coloane (`1≤m,n≤50`) având componente cifre zecimale. Fiecare linie a tabloului reprezintă cifrele câte unui număr natural. Se cere să se afișeze cifrele sumei celor `m` numere date prin tabloul anterior.

Exemplu. Dacă `A` = $\begin{pmatrix} 9 & 0 & 9 & 8 & 3 & 4 & 9 \\ 3 & 5 & 9 & 9 & 8 & 7 & 9 \\ 4 & 9 & 3 & 6 & 7 & 9 & 0 \end{pmatrix}$ se va afișa valoarea `17635018`.

223. Se dă o matrice pătratică de ordinul n. Se consideră că diagonalele împart matricea în 4 zone: nord, sud, est şi vest (elementele de pe diagonală nu fac parte din nici o zonă).

a) Să se calculeze suma elementelor din nord
b) Să se calculeze produsul elementelor din sud.
c) Să se calculeze media aritmetică a elementelor din est.
d) Să se calculeze numărul elementelor nule din vest.
e) Să se obţină simetrica matricei iniţiale faţă de diagonala principală.
f) Să se obţină simetrica matricei iniţiale faţă de diagonala secundară.
g) Să se obţină simetrica matricei iniţiale faţă de o axă orizontală ce trece prin centrul matricei.
h) Să se obţină simetrica matricei iniţiale faţă de o axă verticală ce trece prin centrul matricei.

Exemplu. Pentru matricea $A = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 0 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{pmatrix}$ se obţine

a) 5; b) 182; c) 6.5; d) 1; e) $\begin{pmatrix} 1 & 5 & 8 & 12 \\ 2 & 6 & 9 & 13 \\ 3 & 7 & 10 & 14 \\ 4 & 0 & 11 & 15 \end{pmatrix}$; f) $\begin{pmatrix} 15 & 11 & 0 & 4 \\ 14 & 10 & 7 & 3 \\ 13 & 9 & 6 & 2 \\ 12 & 8 & 5 & 1 \end{pmatrix}$;

g) $\begin{pmatrix} 12 & 13 & 14 & 15 \\ 8 & 9 & 10 & 11 \\ 5 & 6 & 7 & 0 \\ 1 & 2 & 3 & 4 \end{pmatrix}$; h) $\begin{pmatrix} 4 & 3 & 2 & 1 \\ 0 & 7 & 6 & 5 \\ 11 & 10 & 9 & 8 \\ 15 & 14 & 13 & 12 \end{pmatrix}$.

224. Se dă un şir de numere întregi x de dimensiune $N \times N$. Să se genereze o matrice $A \in M(N \times N)$ astfel încât elementele matricei să reprezinte elementele vectorului x scrise în următoarea ordine:

x[1] x[2] x[4] x[7]
x[3] x[5] x[8]
x[6] x[9]
x[10]

Exemplu. Pentru n=3 şi x=(1,5,3,9,4,7,6,10,3) se obţine matricea

$$\begin{pmatrix} 1 & 5 & 9 \\ 3 & 4 & 6 \\ 7 & 10 & 3 \end{pmatrix}.$$

225. Se dă un şir X de numere întregi de dimensiune N × N. Să se genereze o matrice A ∈ M(N × N) astfel încât elementele matricei să reprezinte elementele vectorului X scrise în următoarea ordine:

x[1] x[2] x[5]
x[4] x[3] x[6]
x[9] x[8] x[7]

Exemplu. Pentru n=3 şi x=(1,5,3,9,4,7,6,10,3) se obţine matricea

$$\begin{pmatrix} 1 & 5 & 4 \\ 9 & 3 & 7 \\ 3 & 10 & 6 \end{pmatrix}.$$

226. Să se scrie un program care să dea reprezentarea stelelor unei porţiuni de cer, noaptea. Intrarea va conţine un tablou A care dă digitizarea unei fotografii şi ale cărui elemente reprezintă luminozitatea acelei porţiuni de imagine când a fost făcută fotografia. Luminozităţile iau valori între 0 şi 20.

La ieşire, o stea este marcată cu * pe poziţia (i,j) dacă această poziţie din tablou verifică relaţia:

(A(i,j)+suma luminozităţilor vecine)/5 > 6.

Exemplu. Pentru matricea $\begin{pmatrix} 0 & 0 & 14 & 6 \\ 2 & 2 & 0 & 4 \\ 0 & 8 & 3 & 6 \\ 3 & 8 & 7 & 4 \end{pmatrix}$ se va afişa următoarea "hartă"

$$\begin{pmatrix} . & & & \\ . & & * & * \\ . & * & * & \\ . & & * & . \end{pmatrix}$$

227. Dându-se o matrice pătrată de dimensiune n*n ce conţine toate valorile de la 1 la n^2, se cere să se verifice dacă matricea respectivă reprezintă sau nu un traseu corect de acoperire a tablei de şah prin deplasări ale calului (numerele din matrice reprezintă numărul paşului la care calul se află în căsuţa respectivă).

Exemplu. Matricea $\begin{pmatrix} 1 & 18 & 5 & 10 & 3 \\ 6 & 11 & 2 & 19 & 14 \\ 17 & 22 & 13 & 4 & 9 \\ 12 & 7 & 24 & 15 & 20 \\ 23 & 16 & 21 & 8 & 25 \end{pmatrix}$ corespunde unui traseu corect de acoperire a tablei de şah de dimensiune 5*5 prin deplasări ale calului.

228. Fie a matrice A cu m linii şi n coloane reprezentând o populaţie de viruşi care evoluează astfel:

a) un virus moare când are mai puţin de doi vecini (de plictiseală) sau mai mult de 3 vecini (sufocat);

b) un virus apare într-un loc liber dacă are 2 vecini sau 3 vecini;

c) un virus trăieşte cel mult 3 generaţii, după care moare de bătrâneţe.

Să se afişeze configuraţia populaţiei de viruşi după k generaţii.

Exemplu. Pentru m=6, n=7, k=4

prima generaţie	generaţia a IV-a va avea configuraţia
* – * * * – *	* * – – – – *
* – * – – * –	– – – – – – –
– * * – * * –	* – – * – * *
* * * * * * *	– – * * * – –
* – * * – * –	– – * – * * –
– * * * – * *	* – * – * – *

229. O rază de lumină porneşte din colţul din stânga sus al unui chenar de dimensiune m×n, formând un unghi de 45° cu muchiile. Raza se reflectă când ajunge pe prima sau ultima linie, respectiv prima sau ultima coloană. Să se afişeze toate poziţiile atinse până când raza ajunge într-un colţ al chenarului.

Exemplu. Dacă m=4 şi n=8 atunci parcursul razei este următorul

```
 1      13      19      7
    2      12      6      8
15      3       5      9
   16      4      10     22
```

şi deci ajunge în colţul dreapta jos.

230. Să se simuleze un joc de BINGO cu calculatorul. O tablă de dimensiune n×n se completează p poziţii generate aleator cu numere întregi strict pozitive citite de la tastatură. Jucătorul – care nu ştie cum sunt aranjate numerele în tabelă – trebuie să introducă un număr de m ($2 \le m \le n$) numere din cele date în tabelă şi câştigă dacă din cele m numere au fost extrase toate elementele unei linii sau a unei coloane a tabelei.

Exemplu. Pentru n=5, p=25, m=10, numerele din tabelă 1, 2, ..., 25 iar numerele extrase 2, 3, 5, 10, 15, 16, 17, 18, 20, 23 s-a obţinut (aleator! deci la fiecare rulare tabela va fi alta) tabela $\begin{pmatrix} 1 & 8 & 6 & 18 & 7 \\ 4 & 22 & 5 & 3 & 20 \\ 12 & 11 & 19 & 23 & 13 \\ 21 & 9 & 16 & 15 & 10 \\ 14 & 2 & 24 & 17 & 25 \end{pmatrix}$ deci jucătorul a câştigat (a fost completată coloana a 4-a).

231. Două matrice `A`, `B` pătrate de ordinul `n`, `A` simetrică față de diagonala principală și `B` simetrică față de diagonala secundară, sunt memorate în calculator sub forma a doi vectori `X` și respectiv `Y` conținând elementele de deasupra diagonalei principale, respectiv secundare, inclusiv diagonalele, luate pe linii. Se cere:

a) citind elementele matricelor `A` și `B` să se afișeze vectorii `X` și `Y`;

Exemplu. Pentru `n=4`, $A = \begin{pmatrix} 5 & 1 & 8 & 3 \\ 1 & 4 & 2 & 11 \\ 8 & 2 & 6 & -2 \\ 3 & 11 & -2 & 9 \end{pmatrix}$ și $B = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 3 \\ 8 & 9 & 6 & 2 \\ 10 & 8 & 5 & 1 \end{pmatrix}$ se obțin vectorii `x=(5,1,8,3,4,2,11,6,-2,9)` și `y=(1,2,3,4,5,6,7,8,9,10)`.

b) folosind vectorii `X` și `Y` să se calculeze suma celor două matrice;

Exemplu. Pentru `n=5`, `x=(1,2,3,4,5,6,7,8,9,10,11,12,13,14,15)` și `y=(5,2,3,1,6,0,1,4,3,7,-1,2,-5,8,10)` se va obține matricea

$$A = \begin{pmatrix} 6 & 4 & 6 & 5 & 11 \\ 2 & 7 & 11 & 11 & 10 \\ 10 & 6 & 12 & 15 & 15 \\ -1 & 16 & 10 & 14 & 16 \\ 15 & 4 & 19 & 14 & 20 \end{pmatrix}.$$

c) folosind vectorii `X` și `Y` să se calculeze produsul celor două matrice.

Exemplu. Pentru `n=4` și `x=(5,1,8,3,4,2,11,6,-2,9)` și `y=(1,2,3,4,5,6,7,8,9,10)` se obține matricea $A = \begin{pmatrix} 104 & 112 & 85 & 42 \\ 147 & 132 & 98 & 31 \\ 46 & 66 & 64 & 48 \\ 132 & 126 & 119 & 50 \end{pmatrix}$.

Observație. La punctele b) și c) nu se vor reconstitui cele două matrice ca apoi să se lucreze cu acestea!

232. Fiind dat un cub de dimensiune `n` în care elementele sunt numere de la `1` la n^3, să se determine elementele care sunt așezate pe cele patru diagonale și suma elementelor de pe fiecare diagonală.

Exemplu. Dacă numerotăm "straturile" cubului de sus în jos începând cu `1` iar valorile din fiecare strat sunt:

stratul 1: $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$; stratul 2: $\begin{pmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{pmatrix}$; stratul 3: $\begin{pmatrix} 19 & 20 & 21 \\ 22 & 23 & 24 \\ 25 & 26 & 27 \end{pmatrix}$

atunci cele patru diagonale sunt: diagonala 1: 1, 14, 27 (suma 42); diagonala 2: 3, 14, 25 (suma 42); diagonala 3: 9, 14, 19 (suma 42); diagonala 4: 7, 14, 21 (suma 42).

233. O *bandă generalizată* M(n,a,b) este o matrice pătratică M(n,n) în care toate elementele nenule se află într-o bandă care cuprinde diagonala principală, a-1 diagonale deasupra diagonalei prineipale şi b-1 diagonale sub cea principală. Ea se poate memora într-un vector x care cuprinde diagonalele, începând cu cea mai de sus şi până la cea mai de jos. Să se scrie programul de construire a vectorului x, când se dă matricea M(n,a,b).

Observaţie. Şi între elementele din bandă pot exista elemente nule! Numerele a şi b nu se vor citi de la tastatură ci se determină prin program!

Exemplu. Pentru matricea

$$M = \begin{pmatrix} 1 & 2 & 0 & 0 & 0 & 0 \\ 0 & 5 & 1 & 19 & 0 & 0 \\ 0 & 12 & 8 & 4 & 5 & 0 \\ 0 & 0 & 9 & 3 & 8 & 0 \\ 0 & 0 & 0 & 25 & 2 & 3 \\ 0 & 0 & 0 & 0 & 14 & 6 \end{pmatrix}$$

se obţine vectorul x=(0,19,5,0,2,1,4,8,3,1,5,8,3,2,6,0,12,9, 25,14).

234. Imaginea codificată a unui perete avariat se reprezintă sub forma unei matrice cu m linii şi n coloane cu elemente 0 şi 1, 1 reprezentând perete intact, iar 0 perete avariat. Refacerea acestui perete se face prin plasarea pe verticală a unor blocuri de înălţimi k, k=1,..,m, şi lăţime 1, în locurile avariate.

Pentru o structură dată a unui perete se cere să se determine numărul minim al blocurilor, de înălţimi k=1, k=2, ..., k=m, necesare refacerii peretelui.

Exemplu. Pentru m=5, n=10 şi următoarea structură a peretelui

1110000111
1100001111
1000000011
1111101111
1110000111

se vor folosi 7 blocuri de înălţime 1, 1 bloc de înălţime 2, 2 blocuri de înălţime 3 şi 1 bloc de înălţime 5.

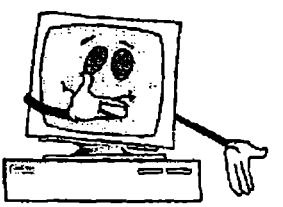
235. Pe o tablă de şah de dimensiune n×n se află un număr oarecare de pioni, fiecare într-un pătrat separat. Vecinii unui pion se află pe una din cele 3 direcţii. Pionii sunt aranjaţi astfel încât formează o singură figură (fără goluri) care poate ocupa şi pătratele de pe marginea pătratului. Să se determine numărul de pioni de pe conturul figurii formate, precum şi numărul total de

pioni cu care s-a format figura. Un pion se consideră că este pe contur dacă numărul vecinilor este strict mai decât 8.

Exemplu. Pentru tabla din figura următoare, în care am marcat cu * pionii, figura este formată din 20 pioni din care doar 2 nu se găsesc pe contur şi anume (3,5) şi (4,5)

```
  * *
* * * * *
*   * * *
  * * * *
  * * * *
    *
```

3.2.2. Maxime şi minime



236. Se dă o matrice A cu m linii şi n coloane cu elemente 0 şi 1. Să se scrie un program care afişează numărul maxim de elemente egale cu 1 conţinute pe o linie.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}$ se va afişa numărul 4.

237. Se dă o matrice cu m linii şi n coloane ($1 \le m, n \le 50$) având componente numere întregi. Se cere să se afişeze numărul/numerele de ordine a liniilor din tablou care au cele mai multe componente distincte.

Exemplu. Pentru m=4, n=5 şi matricea $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 7 & 5 & 6 & 5 & 5 \\ 8 & 9 & 8 & 4 & 4 \\ 6 & 7 & 6 & 9 & 6 \end{pmatrix}$ se va afişa 1.

238. Să se afişeze maximul dintre elementele pare ale liniei L a unei matrice cu m linii şi n coloane. Valorile lui n şi L, precum şi elementele matricei (numere întregi) se citesc de la tastatură.

Exemplu. Pentru m=4, n=4, L=3 şi matricea $\begin{pmatrix} 2 & 3 & 9 & 4 \\ 5 & 3 & 8 & 4 \\ 2 & 6 & 31 & 4 \\ 1 & 3 & 54 & 9 \end{pmatrix}$ se va afişa valoarea 6.

239. Să se scrie un program care însumează elementele maxime de pe fiecare linie a unei matrice `A` cu `m` linii şi `n` coloane.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 9 & 3 & 4 & 7 \\ 9 & 6 & 12 & 5 & 10 \\ 5 & 2 & 7 & 1 & 3 \end{pmatrix}$ suma este `28`.

240. Se citeşte de la tastatură o matrice `a` cu `m` linii şi `n` coloane având elemente numere întregi. Să se determine linia (liniile) din matrice care conţine cele mai multe elemente nenule.

Exemplu. În matricea $\begin{pmatrix} 2 & 0 & 4 & 2 & 0 \\ 6 & 4 & 0 & 3 & 7 \\ 0 & 2 & 0 & 7 & 0 \end{pmatrix}$ linia a doua are cel mai mare număr de elemente nenule.

241. Se dă un tablou bidimensional `a` cu `n` linii şi `n` coloane având componente binare care codifică relaţiile existente între elevii unei şcoli. Un element `a[i,j]` este `0` dacă `i` nu-l cunoaşte pe `j`, `a[i,j]` este `1` dacă `i` cunoaşte pe `j`. Se ştie că aceste relaţii nu sunt simetrice (dacă `i` îl cunoaşte pe `j` nu înseamnă că şi `j` îl cunoaşte pe `i`). Se cere să se afişeze:

a) elevul (elevii) cel mai popular din şcoală, adică elevul (elevii) cunoscut de cei mai mulţi dintre colegi;

b) elevul (elevii) care cunoaşte pe cei mai mulţi dintre colegii săi;

c) perechile de elevi `(i,j)` cu proprietatea că atât `i` îl cunoaşte pe `j` cât şi `j` îl cunoaşte pe `i`.

Exemplu. Dacă relaţiile se codifică prin următoarea matrice $\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$ atunci se va afişa: a) `2,5`; b) `3,4`; c) `(2,4)`, `(2,5)`, `(3,5)`.

242. Se dă o matrice de numere întregi cu `m` linii şi `n` coloane (`1≤m,n≤50`). Să se determine cel mai mare număr de pe marginea matricei, precum şi de câte ori apare el.

Exemplu. Pentru matricea $\begin{pmatrix} 5 & 19 & 6 & 3 \\ 3 & 54 & 18 & 19 \\ 8 & 39 & 64 & 10 \\ 15 & 4 & 7 & 2 \end{pmatrix}$ elementul maxim de pe contur este `19` şi apare de două ori.

243. Se dă un tablou bidimensional `a` cu `n` linii şi `n` coloane având componente binare care codifică relaţiile de prietenie între elevii unei clase. Un element `a[i,j]` este `1` dacă `i` şi `j` sunt prieteni. Se cere să se afişeze copiii care au cei mai mulţi prieteni, ştiind că relaţia de prietenie este simetrică.

Exemplu. Dacă relaţiile se codifică prin următoarea matrice $\begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}$

atunci elevii `1` şi `5` au cei mai mulţi prieteni (`3` prieteni).

244. Se dă o matrice `a` cu `m` linii şi `n` coloane reprezentând o populaţie, $a_{ij}=1$ reprezintă o căsuţă ocupară de un membru al populaţiei, iar $a_{ij}=0$ o căsuţă liberă. Să se determine factorul de aglomerare al populaţiei, care se defineşte ca fiind numărul maxim de vecini ai unui membru al populaţiei.

Exemplu. Pentru următoarea structură a populaţiei

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \end{pmatrix}$$

gradul de aglomerare este `7`.

245. Fie `A` o matrice cu `m` linii şi `n` coloane, cu numere reale. Să se determine linia `l` şi coloana `k` pe care suma elementelor este maximă.

Exemplu. Pentru matricea $\begin{pmatrix} 6 & 2 & 8 & 4 & 7 \\ 7 & 4 & 9 & 14 & 5 \\ -5 & -10 & 0 & 2 & 3 \\ 8 & -2 & 34 & 17 & 10 \end{pmatrix}$ se obţine `l=4` şi `k=3`.

246. O matrice pătrată `A` conţine doar elemente `0` şi `1`. Să se afişeze poziţiile celor mai îndepărtate două elemente de `1` (între ele aflându-se doar valori de `0`), distanţa dintre două elemente fiind calculată ca diferenţa numerelor de ordine ale acestor elemente obţinute prin liniarizarea pe linii a matricei. (Nu se va realiza efectiv liniarizarea matricei.)

Exemplu. În matricea $\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$ cele mai îndepărtate elemente de `1` sunt `a[3,5]` şi `a[5,2]` între ele găsindu-se `8` elemente de `0`.

247. O persoană are de cumpărat din m magazine n produse care au prețuri diferite. Să se scrie un program care să indice pentru fiecare produs magazinul care are prețul minim. Cunoscând cantitățile ce trebuie cumpărate din fiecare produs, să se determine suma ce urmează a fi cheltuită.

Exemplu. Pentru n=3 produse, m=4 magazine și prețurile:

produsul 1: 1500, 2000, 1700, 2400;

produsul 2: 18000, 14500, 16000, 15000;

produsul 3: 33000, 45000, 33000, 33000

și cantitățile 30, 10, 5, se vor afișa următoarele:

Magazinul 1 are prețul minim la produsul 1

Magazinul 2 are prețul minim la produsul 2

Magazinul 1 are prețul minim la produsul 3

Magazinul 3 are prețul minim la produsul 3

Magazinul 4 are prețul minim la produsul 3

Suma totală necesară este 355000.

248. Să se scrie un program pentru determinarea elementului maxim dintr-o matrice dreptunghiulară A de dimensiune mxn, și a tuturor pozițiilor sale.

Exemplu. În matricea $\begin{pmatrix} 5 & 46 & 3 & 1 & 15 \\ 19 & 38 & 7 & 46 & 2 \\ 6 & 9 & 5 & 4 & 20 \\ -55 & 13 & 44 & 0 & 2 \end{pmatrix}$ elementul maxim este 46 și apare în pozițiile (1,2) și respectiv (2,4).

249. Să se calculeze norma matricei A cu m linii și n coloane folosind formula

$$\|A\| = \max\left\{\sum_{j=1}^{n} |a_{ij}|,\ i = 1,2,\ldots,m\right\}.$$

Numerele m și n precum și elementele matricei se citesc de la tastatură.

Exemplu. Norma matricei $A = \begin{pmatrix} -14 & 2 & 24 & -17 \\ 12 & 11 & -19 & 23 \\ 21 & -9 & 16 & 15 \end{pmatrix}$ este 65.

250. Să se scrie un program pentru determinarea elementului maxim din minimele de pe fiecare linie a unei matrice A cu m linii și n coloane.

Exemplu. În matricea $\begin{pmatrix} 6 & 2 & 8 & 4 & 7 \\ 7 & 4 & 9 & 14 & 5 \\ -5 & -10 & 0 & 2 & 3 \\ 8 & -2 & 34 & 17 & 10 \end{pmatrix}$ elementul cerut este 4.

251. Să se determine toate punctele şa ale unei matrice A (de dimensiune mxn) şi poziţiile acestor puncte şa. Elementul a_{ij} este punct şa a matricei dacă este minim pe linia i şi maxim pe coloana j sau invers.

Exemplu. În matricea $\begin{pmatrix} 6 & 2 & 8 & 4 & 7 \\ 7 & 4 & 9 & 14 & 5 \\ -5 & -10 & 0 & 2 & 3 \\ 8 & -2 & 34 & 17 & 10 \end{pmatrix}$ se vor afişa valorile a[2,2]=4 şi a[3,5]=3.

3.2.3. Căutări de elemente

252. Să se cripteze un text alfanumeric (conţinând litere mari, spaţiu, virgulă, punct şi asterisc) cu ajutorul următoarei matrice de criptare:

V U L P E J
A B C D F
G H I K M
N O Q R S
T W X Y Z *

criptarea realizându-se pe grupe de 2 caractere, prin înlocuirea caracterelor succesive m_1, m_2 din text cu perechea de caractere c_1, c_2 determinată conform regulilor:

a) dacă m_1, m_2 apar ca vârfuri opuse ale unui dreptunghi în cadrul matricei m_1 şi m_2 vor fi înlocuite cu caracterele corespunzătoare celorlalte două vârfuri ale dreptunghiului, în sens invers trigonometric (de exemplu BR→DO, RC→QD);

b) dacă m_1, m_2 apar pe aceeaşi linie, c_1 şi c_2 se obţin printr-o deplasare ciclică la dreapta (de exemplu AD→BF, O, →QN);

c) dacă m1 şi m2 se găsesc pe aceeaşi coloană, c_1 şi c_2 se obţin printr-o deplasare ciclică în jos (HW->OU);

d) dacă m_1, m_2 coincid, ele rămân nemodificate în textul criptat;

e) dacă textul conţine un număr impar de caractere, ultimul se grupează cu blancul.

Exemplu. Textul '**ACESTA ESTE UN EXEMPLU SIMPLU. **' se codifică astfel '**BDFZVGFJZNJFOVFJLZKEPL,FK.EPJH.JJ.'

253. Fiind dată o matrice cu m linii şi n coloane cu elemente numere întregi, scrieţi un program care stabileşte de câte ori apare o valoare dată x ca element al matricii;

Exemplu. Pentru m=4, n=6, x=3 şi matricea

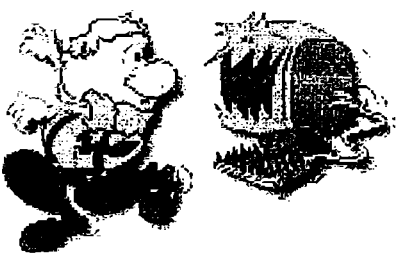
$$\begin{pmatrix} 1 & 9 & 3 & 5 & 2 & 6 \\ 4 & 2 & 3 & 1 & 5 & 9 \\ 8 & 7 & 1 & 0 & 3 & 3 \\ 10 & 6 & 4 & 1 & 98 & 6 \end{pmatrix}$$

se afişează valoarea 4.

254. Să se verifice dacă există o valoare comună tuturor liniilor, respectiv coloanelor unei matrice A cu m linii şi n coloane.

Exemplu. În matricea $\begin{pmatrix} 1 & 4 & 5 & 6 \\ 1 & 7 & 8 & 9 \\ 3 & 10 & 1 & 3 \\ 11 & 3 & 3 & 1 \end{pmatrix}$ numărul 3 este comun tuturor coloanelor iar numărul 1 este comun tuturor liniilor.

3.2.4. Sortări şi interclasări



255. Se dă o matrice A de numere reale cu M linii şi N coloane. Să se schimbe între ele liniile matricei A astfel încât prima coloană să devină ordonată crescător.

Exemplu. Matricea $\begin{pmatrix} 4 & 5 & 4 \\ 1 & 9 & 7 \\ 2 & 4 & 3 \end{pmatrix}$ devine după sortare $\begin{pmatrix} 1 & 9 & 7 \\ 2 & 4 & 3 \\ 4 & 5 & 4 \end{pmatrix}$.

256. Se consideră o matrice A de dimensiune nxm, n≤100, m≤100. Să se reaşeze elementele în matrice astfel încât ele să apară în ordine crescătoare atât pe linii cât şi pe coloane.

Exemplu. Dacă matricea iniţială este $\begin{pmatrix} 2 & 98 & 36 \\ 1 & 14 & 7 \\ 55 & 18 & 12 \end{pmatrix}$ se poate obţine matricea $\begin{pmatrix} 1 & 7 & 14 \\ 2 & 18 & 55 \\ 12 & 36 & 98 \end{pmatrix}$ (nu este unica soluţie!).

257. Se citeşte de la tastatură o matrice cu elemente reale $A = (a_{ij})_{\substack{i=\overline{1,m}\\ j=\overline{1,n}}}$. Să se schimbe între ele liniile matricei astfel încât şirul sumelor $s_1, s_2, \ldots, s_m$ să fie ordonat descrescător, unde

$$s_i = \sum_{j=1}^{n} a_{ij}, \quad i = \overline{1,m}.$$

Exemplu. Matricea $\begin{pmatrix} 1 & 5 & 4 \\ 2 & 9 & 7 \\ 6 & 4 & 3 \end{pmatrix}$ devine după sortare $\begin{pmatrix} 2 & 9 & 7 \\ 6 & 4 & 3 \\ 1 & 5 & 4 \end{pmatrix}$.

258. Să se aranjeze elementele unei matrice pătrate, astfel încât parcurgând matricea în spirală (dinspre exterior spre interior) să se obţină elementele în ordine crescătoare. Nu se vor folosi alte date structurate.

Exemplu. După sortarea matricei $\begin{pmatrix} 1 & 8 & 6 & 18 \\ 4 & 122 & 5 & 3 \\ 12 & 11 & 19 & 23 \\ 21 & 9 & 16 & 15 \end{pmatrix}$ găsim matricea $\begin{pmatrix} 1 & 3 & 4 & 5 \\ 18 & 19 & 21 & 6 \\ 16 & 122 & 23 & 8 \\ 15 & 12 & 11 & 9 \end{pmatrix}$.

259. Să se aranjeze elementele unei matrice pătratice astfel încât parcurgând matricea pe diagonale paralele cu diagonala principală începând cu cea mai de sus diagonală şi dinspre stânga spre dreapta să se obţină elementele în ordine crescătoare.

Exemplu. După sortarea matricei $\begin{pmatrix} 1 & 8 & 6 & 18 \\ 4 & 122 & 5 & 3 \\ 12 & 11 & 19 & 23 \\ 21 & 9 & 16 & 15 \end{pmatrix}$ găsim matricea $\begin{pmatrix} 9 & 5 & 3 & 1 \\ 16 & 11 & 6 & 4 \\ 21 & 18 & 12 & 8 \\ 122 & 23 & 19 & 15 \end{pmatrix}$.

260. Se consideră o matrice rară (adică cu foarte multe elemente nule) `A(n,n)` cu `p` elemente nenule. Matricea este memorată economic sub forma a doi vectori `V(i)`, `T(i)`, $1 \le i \le p$, care reţin valorile elementelor nenule ale matricei, respectiv poziţia acestor elemente, liniarizate pe linii.

Să se scrie **câte** un program pentru fiecare din următoarele operaţii:

a) citind vectorii V şi T reface matricea A;

Exemplu. Pentru n=5, p=8, V=(2,7,8,4,6,1,3,5), T=(1,5,9,11,14, 18,20,22) se va afişa matricea

$$A = \begin{pmatrix} 2 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 8 & 0 \\ 4 & 0 & 0 & 6 & 0 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 5 & 0 & 0 & 0 \end{pmatrix}.$$

b) determină vectorii V şi T corespunzători matricei transpuse a matricei A, fără a reconstitui matricea;

Exemplu. Transpusa matricei anterioare este dată prin n=5, V=(2,4,5,1,8,6,7,3) şi T=(1,3,10,14,17,18,21,24).

c) considerând matricele A(n,n) şi B(n,n) cu p respectiv q elemente nenule, calculează suma celor două matrice folosind memorarea economică:

Exemplu. Pentru n=5 şi matricele date prin V1=(2,7,8,4,6,1,3,5), T1=(1,5,9,11,14,18,20,22) şi respectiv V2=(1,5,1,3,4,4,2,9), T2=(1,2,8,10,12,16,18,24) se obţine V=(3,5,7,1,8,3,4,4,6,4, 3,3,5,9) şi T=(1,2,5,8,9,10,11,12,14,16,18,20,22,24).

d) acelaşi enunţ ca la punctul anterior pentru produsul celor două matrice.

Exemplu. Pentru n=5 şi matricele date prin V1=(2,7,8,4,6,1,3,5), T1=(1,5,9,11,14,18,20,22) şi respectiv V2=(1,5,1,3,4,4,2,9), T2=(1,2,8,10,12,16,18,24) se obţine V=(2,10,63,32,16,28,20, 12,4,27,5,15) şi T=(1,2,4,6,8,11,12,13,17,19,23,25).

Observaţie. La toate punctele se consideră că elementele nenule se vor citi/afişa în ordinea liniarizării pe linii a matricei (adică în ordinea $a_{11}, a_{12}, \ldots, a_{1n}, a_{21}, \ldots, a_{2n}, \ldots, a_{nn}$, deci vectorul T este sortat crescător).

261. Se consideră o matrice A(n,n) cu p elemente nenule. Matricea este memorată economic sub forma a trei vectori V(i), L(i) şi C(i) care reţin valorile, liniile şi coloanele elementelor nenule.

Să se scrie **câte** un program care:

a) citind vectorii V, L, C reface matricea A;

Exemplu. Pentru n=5, V=(1,5,1,3,4,4,2,9), L=(1,1,2,2,3,4,4,5), C=(1,2,3,5,2,1,3,4) se obţine matricea $\begin{pmatrix} 1 & 5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 4 & 0 & 0 & 0 \\ 4 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 \end{pmatrix}$.

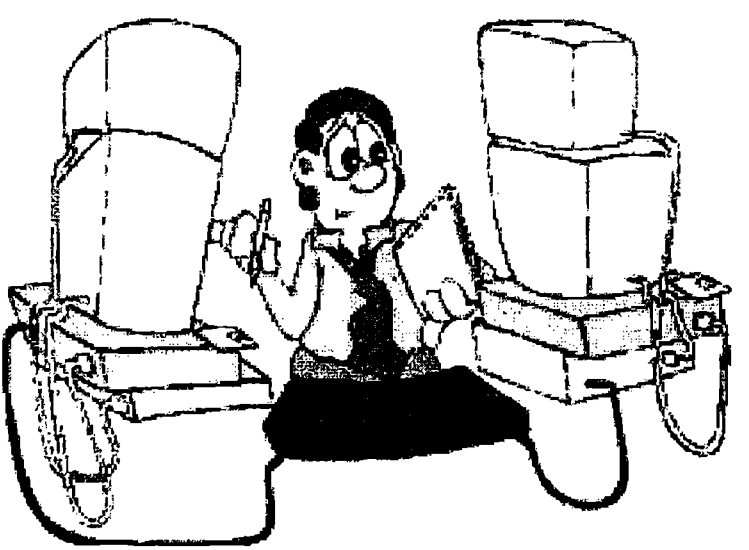
b) considerând matricele `A(n,n)` şi `B(n,n)` cu `p` şi respectiv `q` elemente nenule, calculează suma celor două matrice folosind memorarea economică.

Exemplu. Pentru `n=5` şi `V1=(1,5,1,3,4,4,2,8)`, `L1=(1,1,2,2,3,4,4,5)`, `C1=(1,2,3,5,2,1,3,4)`; respectiv `V2=(2,7,8,4,6,1,3,5)`, `L2=(1,1,2,3,3,4,4,5)`, `C2=(1,5,4,1,4,3,5,2)` se obţin vectorii `V=(3,5,7,1,8,3,4,4,6,4,3,3,5,8)`, `L=(1,1,1,2,2,2,3,3,3,4,4,4,5,5)` şi `C=(1,2,5,3,4,5,1,2,4,1,3,5,2,3)` corespunzători matricei sumă.

c) acelaşi enunţ pentru produsul celor două matrice.

Exemplu. Pentru `n=5` şi `V1=(1,5,1,3,4,4,2,8)`, `L1=(1,1,2,2,3,4,4,5)`, `C1=(1,2,3,5,2,1,3,4)`; respectiv `V2=(2,7,8,4,6,1,3,5)`, `L2=(1,1,2,3,3,4,4,5)`, `C2=(1,5,4,1,4,3,5,2)` se obţin vectorii `V=(2,40,7,4,15,6,32,16,12,28,8,24)`, `L=(1,1,1,2,2,2,3,4,4,4,5,5)` şi `C=(1,4,5,1,2,4,4,1,4,5,3,5)` corespunzători matricei produs.

Observaţie. La toate punctele se consideră că elementele nenule se vor citi/afişa în ordinea liniarizării pe linii a matricei (adică în ordinea a_{11}, a_{12}, ..., a_{1n}, a_{21}, a_{2n}, ... a_{nn}, deci vectorul `T` este sortat crescător).



4. Alte tipuri structurate de date

4.1. Tipul string

1. Să se afişeze toate prefixele şi toate sufixele unui cuvânt citit de la tastatură.

   Exemplu. Prefixele cuvântului `'animal'` sunt: `'a'`, `'an'`, `'ani'`, `'anim'`, `'anima'`, `'animal'` iar sufixele sunt `'animal'`, `'nimal'`, `'imal'`, `'mal'`, `'al'`, `'l'`

2. Se citeşte un şir de caractere (mari şi mici) de la tastatură. Se cere
   a) Să se afişeze şirul de caractere doar cu litere mari.
   b) Să se afişeze şirul de caractere doar cu litere mici.

   Exemplu. şirul `'AbcDEFghIj'` va deveni a) `'ABCDEFGHIJ'` şi respectiv b) `'abcdefghij'`

3. De pe un rând al ecranului se citeşte un text care poate să conţină orice caractere (litere mari, litere mici, cifre şi caractere speciale). Să se tipărească textul obţinut prin transformarea în litere mari a literelor mici aflate pe poziţii impare în cadrul textului dat (prima, a treia, a cincea etc.).

   Exemplu. Şirul `'AbcdeFGHijkl5NOPRstu'` devine `'AbCdEFGHIjKl5NOPRsTu'`.

4. Se citeşte de la tastatură un caracter `c` şi un text de maxim `70` caractere. Textul este citit pe un singur rând de ecran. Afişaţi de câte ori apare caracterul `c` în cadrul textului. Literele mici nu se vor considera diferite de majuscule.

   Exemplu. În textul `"Acesta este un exemplu destul de banal"` litera `'a'` apare de `4` ori.

5. Se dă un şir de caractere `c` şi o mulţime `A` de caractere. Să se determine câte din elementele mulţimii `A` apar în scrierea lui `c`.

   Exemplu. Dacă `c='Elefantul are trompa dar iepurele are urechi lungi'` iar `A=('a','x','g','e','v')` atunci se vor afişa literele `'a'`, `'g'` şi `'e'`.

6. Să se elimine toate aparitiile caracterului spaţiu dintr-un text.

   Exemplu. Dacă textul iniţial este `' abc def ghi '` în final se va obţine `'abcdefghi'`

7. Se citeşte un text de la tastatură. Să se elimine spaţiile inutile (caracterul spaţiu înainte de unul din caracterele . ; , sau mai mult de un spaţiu între două cuvinte, sau spatiile de la începutul şi de la sfârşitul textului). După eliminare să se tipărească textul obţinut.

   Exemplu. Textul `abc def  ghi` devine '`abc def, ghi`'.

8. Să se scrie un program care citeşte mai multe linii de la tastatura şi o tipăreşte pe cea mai lungă. Citirea de la tastatură se termină când s-a introdus o linie goală.

   Exemplu. Dacă liniile citite de la tastatură sunt:

   ```
   Elevul iese la tabla
   El răspunde la întrebările profesorului
   Daca răspunde corect ia nota mare
   ```

   se va afişa cea de-a doua linie care este cea mai lungă.

9. Se citeşte de la tastatură un şir de caractere care poate să conţină numai litere şi cifre. Afişaţi numărul literelor mari, numărul literelor mici şi numărul caracterelor-cifră din text.

   Exemplu. În textul '`S-a născut la Bucuresti in 1874 pe 4 mai. A trait mai bine de 80 de ani.`' Se găsesc `43` de litere mici, `3` litere mari si `7` cifre.

10. Se citesc de la tastatura mai multe linii. Scrieţi un program care să tipărească toate liniile mai scurte de `80` de caractere. Citirea de la tastatură se termină când s-a introdus o linie goală.

11. Să se verifice dacă două cuvinte introduse de la tastatură sunt rime (dacă ultimele `p` caractere ale celor două cuvinte coincid, unde `p` este o valoare dată).

    Exemplu. Dacă `p=2` atunci cuvintele '`indraznit`' şi '`ferit`' sunt rime.

12. Se citeşte de la tastatură o frază de lungime cel mult `70` de caractere. Între cuvintele frazei pot apărea ca separatori numai spaţiul, virgula şi punctul. Să se afişeze unul sub altul cuvintele frazei în ordine alfabetică.

    Exemplu. Pentru fraza "`Acesta este un exemplu simplu, dar bun, de fraza `" se vor afişa cuvintele în următoarea ordine: '`Acesta`', '`bun`', '`dar`', '`de`', '`este`', '`exemplu`', '`fraza`', '`simplu`', '`un`'

13. Se citeşte de la tastatură un text. Se ştie că două cuvinte din text sunt separate prin unul sau mai multe spaţii. Să se afişeze toate cuvintele din text formate din `k` caractere.

    Exemplu. Textul '`Mergeam la plimbare si vorbeam cu prietena mea`' conţine `3` cuvinte de lungime `k=2`.

14. Scrieți un program care convertește un șir de caractere într-un întreg. Conversia se oprește odată cu întâlnirea unui caracter ce nu este cifră.

Exemplu. Dacă șirul este `'123ef34'` se va obține numărul întreg `123`.

15. Se citește de la tastatură o frază de maxim `70` de caractere. Să se afișeze, una sub alta, toate perechile de vocale consecutive din frază, precum și numărul acestora.

Exemplu. Pentru textul `'aceasta bluza desi foarte ieftina era foarte frumoasa asa ca am cumparat-o'`, are `5` perechi de vocale consecutive și anume: `'ea'`, `'oa'`, `'ie'`, `'oa'`, `'oa'`

16. Se citește un text `T` de la tastatură și două cuvinte `c1` și `c2`. Să se scrie un program care tipărește textul obținut din `T` prin înlocuirea aparițiilor cuvântului `c1` cu cuvântul `c2`. Cuvintele se consideră separate prin unul sau mai multe caractere , ; și spațiu.

Exemplu. Dacă în textul `'Ea este elevă. El este student si mi-a dat o veste foarte buna'` se înlocuiește cuvântul `'este'` cu `'era'` se obține textul `'Ea era elevă. El era student și mi-a dat o veste foarte bună.`

17. Se dau două șiruri de caractere de lungimi egale. Al doilea șir conține doar cifre, în caz contrar programul terminându-se fără nici o prelucrare. Se cere să se construiască un al treilea șir, prin repetarea pe rând, a fiecărui caracter din primul șir, de un număr de ori egal cu cifra corespunzătoare din al doilea șir.

Exemplu. `'abc'` și `'243'` ⇒ `'aabbbbccc'`.

18. Un cuvânt este palindrom dacă citind literele de la dreapta la stânga obținem același cuvânt. Scrieți un program care verifică dacă un cuvânt citit de la tastatură este palindrom sau nu.

Exemplu. Cuvintele `'cojoc'` și 'sas' sunt palindroame.

19. Se citește de la tastatură un cuvânt de lungime cel mult `20` de caractere, format numai din litere mari. Să se afișeze toate cuvintele distincte ce se pot forma prin eliminarea câte unui singur caracter din cuvântul dat.

Exemplu: pentru cuvântul `ELEVII` se vor afișa, nu neapărat în această ordine cuvintele:

`LEVII` `EEVII` `ELVII` `ELEII` `ELEVI`

20. Se citește de la tastatură un text de lungime cel mult `70` caractere alcătuit din cuvinte separate între ele printr-unul sau mai multe spații. Să se afișeze cuvântul (cuvintele) de lungime maximă din text.

Exemplu. În textul `Era toamna si ploua foarte tare` cuvintele de lungime maximă sunt `'toamna'` și `'foarte'`

21. Să se scrie un program care citeşte un text de la tastatură şi îl traduce în limba păsărească astfel: fiecare vocală α va fi înlocuită prin şirul $\alpha p \alpha$.

Exemplu. Pentru şirul '`exemplu de problema`' obţinem '`epexepemplupu depe propoblepemapa`'.

22. Se citesc de la tastatură n şiruri dc caractere care pot conţine numai litere şi cifre. Să se afişeze câte dintre şirurile citite nu pot fi transformate în numere întregi, precum şi suma numerelor rezultate prin transformarea şirurilor care pot fi convertite.

Exemplu. Dacă şirurile citite sunt (`'451','2.3','254','cdg','22','h','10'`) atunci 3 şiruri nu pot fi transformate în numere întregi iar suma numerelor rezultate prin transformarea şirurilor care pot fi convertite este 737.

23. Să se scrie un program care tipăreşte distribuţia frecvenţelor lungimii cuvintelor aflate într-un text citit de la tastatură. Cuvintele sunt separate prin spaţii.

Exemplu. Pentru textul '`Toamna aceasta ploua foarte tare desi nu prea imi place acest lucru`' se va afişa:

1 cuvant de lungime 2
1 cuvant de lungime 3
3 cuvinte de lungime 4
4 cuvinte de lungime 5
2 cuvinte de lungime 6
1 cuvant de lungime 7

24. Să se scrie un program care va realiza începerea fiecărei propoziţii dintr-un text cu literă mare. Spaţiile inutile din text se vor elimina. Dacă nu s-a făcut nici o modificare se va da un mesaj corespunzător. Propoziţiile sunt despărţite între ele de un caracter punct urmat de zero, unul sau mai multe spaţii.

Exemplu. Textul '`toamna a venit devreme. ploua si e frig. soarele nu mai arde aşa de tare.`' se transformă în '`Toamna a venit devreme. Ploua si e frig. Soarele nu mai arde aşa de tare.`'

25. Să se scrie un program care inversează într-un text toate cuvintele care încep cu litera '`a`' (sau '`A`') prin oglindire. Cuvintele din text se consideră separate prin spaţii (unul sau mai multe).

Exemplu. Textul '`Aceasta este un exemplu aproape perfect asa ca am insistat sa-l vezi acum si tu.`' Se va transforma în '`atsaecA este un exemplu epaorpa perfect asa ca ma insistat sa-l vezi muca si tu.`'

26. Să se determine numărul cuvintelor dintr-un text dat care sunt mai lungi de 7 litere şi conţin litera u. Se presupune că în textul dat cuvintele sunt alcătuite numai din litere şi sunt separate prin spaţiu sau virgulă.

Exemplu. Dacă textul este '`Exemplul acesta nu este cu nimic superior, comparativ cu celelalte, este doar un exemplu`' se va afişa valoarea 3.

27. Se dă un şir de caractere. Să se elimine din acesta tot ce nu e literă sau cifră şi să se afişeze în noua formă împreună cu lungimea anterioară şi cea curentă.

Exemplu. Daeă şirul iniţial este '`#exemplu de 3 siruri ^^nice^^ !!! Nota 10`' se obţine şirul '`exemplude3sirurinіceNota10`' Lungimea şirului iniţial este `41` iar a celui final `26`.

28. Să se scrie un algoritm care multiplică un şir de caractere dat de `n` ori.

Exemplu. Fiind dat şirul '`WAKA`' şi `n=2`, algoritmul va genera şirul '`WAKAWAKA`'.

29. Se citesc de la tastatură două şiruri de litere ale alfabetului englez. Să se tipărească vocalele care nu apar în nici unul din cele două şiruri. Un caracter va fi afişat doar o dată. Literele mici nu se vor considera distincte de majuscule.

Exemplu. Dacă cele două şiruri sunt '`Elicopter`' şi '`TELEFON`' atunci se vor afişa literele '`A`', '`U`'

30. Se citesc de la tastatură două şiruri de litere ale alfabetului englez. Să se tipărească caracterele care nu sunt comune celor două şiruri (adică apar doar într-un singur şir). Un caracter va fi afişat doar o dată. Literele mici nu se vor considera distincte de majuscule. Nu se vor folosi alte tipuri structurate de date.

Exemplu. Dacă cele două şiruri sunt '`Elicopter`' şi '`TELEFON`' atunci se vor afişa literele '`C`', '`F`', '`I`', '`N`', '`P`', '`R`'.

31. Se citesc de la tastatură două cuvinte, fiecare în câte o variabilă de tip `string`. Să se determine literele comune celor două cuvinte. Tipărirea unei litere comune se va face o singură dată. Se va folosi funcţia predefinită `pos`. Nu se vor folosi structuri de date suplimentare, iar majusculele nu se consideră diferite de literele mici.

Exemplu. Dacă cele două cuvinte sunt '`paste`' şi respectiv '`saltea`' atunci literele comune sunt '`A`', '`S`', '`T`'

32. Un text conţine litere mari şi una sau mai multe secvenţe:

'`\Punct`' '`\Excl`' '`\Inter`' '`\Virg`'

Să se rescrie textul înlocuind secvenţele de mai sus cu

'`!`' '`?`'

33. Să se afişeze cuvintele comune din două texte citite de la tastatură. Cuvintele sunt despărţite prin spaţii sau caracterul punct. Un cuvânt se va afişa doar o singură dată. Nu se face distincţie între literele mari şi mici, însă la afişare toate cuvintele se vor afişa cu litere mici. Se presupune că introducerea datelor se face corect adică textele nu conţin alte caractere decât litere, spaţii şi caractere punct. Nu se cere validarea datelor.

Exemplu. Dacă textele sunt `'Tot ce as vrea ar fi sa ma odihnesc putin.'` respectiv `'Pasarea care a venit la geamul meu tot mai vrea cateva graunte.'` atunci cuvintele comune sunt `'tot'`, `'vrea'`.

34. Să se verifice dacă două cuvinte se pot obţine unul din celălalt prin permutarea literelor. Nu se face distincţie între literele mari şi mici ale alfabetului.

Exemplu. Cuvântul `'CUPTOR'` se poate obţine din cuvântul `'CoRuPt'` prin permutarea literelor.

35. Se citeşte de la tastatură un număr şir de caractere. Să se afişeze caracterul (caracterele) care apare (apar) de cele mai multe ori în componenţa şirului. Nu se vor folosi alte tipuri structurate de date cu excepţia tipului `string`.

Exemplu. Dacă şirul este `'akvkbuvkv'`, programul va tipări caracterele `'k'` şi `'v'`, care apar de câte trei ori fiecare.

36. Să se scrie un program care transformă un număr scris în baza `16` (număr memorat într-un şir de caractere) în baza `10`.

Exemplu. Pentru numărul `'14A'` se va afişa `330`.

37. Se introduc de la tastatură două numere naturale memorate în două variabile de tip `string` (numărul de cifre <= `250`). Se cere să se afişeze suma celor două numere şi diferenţa dintre cel mai mare şi cel mai mic număr.

Exemplu. `'123'+'9799'='9922'`; `'123'-'96'='27'`

38. Pentru compactarea mesajelor transmise între două terminale se înlocuieşte orice succesiune de mai multe caractere de acelaşi fel cu un caracter de control (fie el '#'), urmat de caracterul respectiv şi de numărul de repetări succesive. Să se scrie programul de compactare şi decompactare a unui text citit de la tastatură.

Exemplu. Şirul `'caaaabaxxxxxxxxxxx'` se codifică `'c#a4ba#x12'`.

39. Se citeşte de la tastatură un număr întreg cu maxim `10` cifre, sub forma unui şir de cifre. Să se elimine o cifră aleasă astfel încât numărul rămas să aibă cifrele în ordine crescătoare. Dacă sunt mai multe soluţii se vor afişa toate, iar dacă problema nu are nici o soluţie se va tipări un mesaj. Un număr se va afişa o singură dată.

Exemplu. Pentru `m=2435`, poate fi eliminată cifra `3` rămânând numărul `245`, sau cifra `4` rămânând numărul `235`.

40. Se dau două şiruri de caractere. Să se construiască un al treilea, luând, pe rând, din fiecare şir, câte o vocală. În momentul în care se termină un şir, se vor lua vocalele din celălalt şir.

Exemplu. `s1='FRANCOFON'`, `s2='TRANSFORMARILE'` => `s='AAOOOAIE'`

41. Se dau `n` propoziţii formate din caractere ale alfabetului englez (mari şi mici). Ultimele caractere ale fiecărei propoziţii sunt cifre care formează un număr întreg. Se cere să se afişeze aceste propoziţii în ordinea crescătoare a numerelor cu care se termină. La afişare numerele respective vor fi eliminate.

Exemplu. Pentru `n=3` şi propoziţiile '`Elicopterul zboara jos 12`', '`Alina citeste o carte 90`', '`Bunica este batrana si bolnava 5`' se va afişa

'`Bunica este batrana si bolnava`'

'`Elicopterul zboara jos`'

'`Alina citeste o carte`'

42. Se dau două numere `a`, `b` cu cel mult `9` cifre. După fixarea a două cifre, câte una pentru fiecare număr, aceste cifre se interschimbă între ele, obţinându-se alte două numere `a'`, `b'`. Să se determine numerele `a'` şi `b'` astfel încât suma `a'+b'` să fie maximă fie maximă.

Exemplu. Pentru `a=23` şi `b=416` => `a'=63` şi `b'=412`.

43. Se dă un şir de cel mult `255` litere mici ale alfabetului englez. Să se găsească cea mai lungă secvenţă în care literele să fie ordonate alfabetic.

Exemplu. Pentru şirul '`sybegifcabfayewfa`' se va afişa secvenţa '`begi`'

44. Se dă un şir format din cel mult `255` litere mici ale alfabetului englez. Să se găsească cea mai lungă secvenţă de caractere "simetrice", din text, formate din cel puţin două caractere ("simetrice" = au aceeaşi formă citite de la stânga la dreapta şi de la dreapta la stânga).

Exemplu. Pentru şirul '`voabcbabxbabmu`' se va afişa secvenţa '`babxbab`'.

45. Se consideră un şir de cel mult `100` de caractere, format din litere mici ale alfabetului englez. Se numeşte *bâlbă* o secvenţă de caractere care apare în şir de cel puţin două ori una după alta. Să se determine cea mai lungă bâlbă din şirul considerat (ca număr de caractere).

Exemplu. Pentru şirul '`vevevevereririririritata`' se va afişa bâlba '`ririri`'

46. Un şir de caractere se spune că are perioada `k` dacă şirul este format prin concatenarea de mai multe ori a unui şir de lungime `k`. Să se determine cea mai scurtă perioadă pentru un şir dat, precum şi subşirul corespunzător acestuia.

Exemplu. Pentru şirul '`abcabcabcabc`' perioada minimă este `3` (şirul este periodic şi cu perioada `6`).

47. Creaţi un editor de linie simplu, care să permită mişcarea pe orizontală în textul scris folosind tastele cursor, ştergerea caracterelor înapoi (folosind `Backspace`) şi înainte (folosind `Del`) şi inserarea de caractere noi în poziţia cursorului. Încheierea editării se face la apăsarea tastei `Enter`. Pentru validarea operaţiilor făcute la sfârşit se va afişa încă o dată şirul obţinut după editare.

48. Să se afişeze pe verticală `n` cuvinte ordonate alfabetic. Cuvintele se vor citi de la tastatură. Aveţi grijă că '`a`'<'`B`' chiar dacă `ord('a')=97>66=ord('B')`.

Exemplu. `mama, are, mere, foarte, frumoase` => se va afişa

```
a  f  f  m  m
r  o  r  a  e
e  a  u  m  r
   r  m  a  e
   t  o
   e  a
      s
      e
```

49. Se dă un şir `s` de caractere de lungime `n`, format doar din litere mai ale alfabetului englez. Spunem că poziţia `i` din şir este mai slabă decât poziţia `j` dacă şirul $s_i s_{i+1} \ldots s_n s_1 \ldots s_{i-1}$ este mai mic în ordine lexicografică decât $s_j \ldots s_n s_1 \ldots s_{j-1}$. Să se determine pentru şirul dat cea mai slabă poziţie.

Exemplu. Pentru şirul `s='ELICOPTER'` cea mai slabă poziţie este `4` (şirul '`COPTERELI`' este cel mai mic în ordine lexicografică).

50. Algoritmii Marcov sunt sisteme formale pentru manipularea şirurilor. Aceste scheme sunt des utilizate în proiectarea limbajelor de programare şi a altor aplicaţii cu un grad înalt de dificultate.

Strategia generală într-un algoritm Marcov este de a lua şirul de intrare `x` şi dintr-un număr de paşi, se transformă `x` într-un şir de ieşire `y`. Atât `x` cât şi `y` sunt şiruri conţinând caractere dintr-un alfabet (mulţime de caractere) `V`.

Paşii (producţiile) sunt de forma `u` → `w` unde `u` şi `w` sunt şiruri de caractere din `V`. O astfel de producţie este aplicabilă unui şir `z` de caractere din `V` dacă există cel puţin o apariţie a lui `u` în `z`, altfel producţia este neaplicabilă şirului `z`. Dacă producţia este aplicabilă prima apariţie a lui `u` în `z` este înlocuită cu `w`.

Exemplu. Producţia '`ba`'→'`c`' aplicată şirului '`ababab`' produce şirul '`acbab`'. Producţia '`baa`'→'`c`' nu este aplicabilă şirului '`ababab`'.

Producţiile se consideră ordonate (prima este prioritară celorlaltor).

Un algoritm Marcov se termină atunci când nu mai există producţii aplicabile.

Datele se citesc de la tastatură (şirul de intrare, numărul de producţii, producţiile). Să se scrie un program care transformă şirul de intrare folosind procedeul descris mai sus. Se vor afişa toate fazele de transformare a şirului de intrare în şirul de ieşire.

Exemplu. Dacă producţiile sunt: `'ab'->'ef'`, `'ef'->'ad'`, `'ad'->'d'`, `'cd'->'ef'`, `'dg'->'d'` atunci şirul iniţial `'abcdefg'` se transformă succesiv astfel: `'abcdefg'` -> `'efcdefg'` -> `'adcdefg'` -> `'adcdadg'` -> `'dcdadg'` -> `'dcddg'` -> `'defdg'` -> `'daddg'` -> `'dddg'` -> `'ddd'`

51. Dându-se un şir de caractere format din literele mici ale alfabetului englez şi caracterul `$` se cere să se determine un cod asociat acestui şir astfel:

    dacă numărul de litere din şir nu este multiplu de 4 codul este 1;

    dacă numărul de litere este multiplu de 4, codul se obţine astfel: fiecare grup de câte 4 litere consecutive (cu ignorarea caracterelor `$`) formează în ordine vârfurile unui patrulater. O latură a acestui patrulater este dată de distanţa în alfabet a literelor aflate în vârfurile laturii respective. Codul şirului se obţine prin adunarea perimetrelor tuturor patrulaterelor ce se obţin.

Exemplu. Pentru şirul `'ofhth$tehalcmu$y$yop$ewq'` se obţine codul 156.

52. Se citeşte de la tastatură un şir de caractere reprezentând un număr întreg scris în sistemul de numeraţie roman. Să se scrie un program care determină scrierea arabă a numărului respectiv ştiind că cifrele sistemului roman sunt:

    `I` – valoarea 1;
    `V` – valoarea 5;
    `X` – valoarea 10;
    `L` – valoarea 50;
    `C` – valoarea 100;
    `D` – valoarea 500;
    `M` – valoarea 1000.

Pentru reprezentarea numerelor se folosesc următoarele reguli:

1) Mai multe cifre identice într-o succesiune formează un număr egal cu suma cifrelor.
2) Un grup de două cifre, în care prima are valoarea mai mică decât a doua, reprezintă un număr care are valoarea egală cu diferenţa dintre valoarea cifrei a doua şi valoarea primei cifre.
3) Un grup de două cifre în care prima are o valoare mai mare decât a doua, reprezintă un număr care are o valoare egală cu suma dintre valoarea primei cifre şi valoarea celei de a doua.

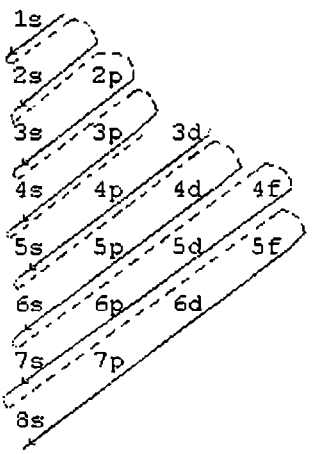
Numerele se citesc însumând grupurile formate după regulile prezentate.

Exemplu. Numărul roman `XCVIII` reprezintă numărul arab 98.

53. Se ştie că pentru caracterizarea structurii şi a proprietăţilor atomilor, un rol deosebit de important îl are configuraţia electronică a atomului respectiv. Reamintim că există mai multe straturi electronice (în prezent se cunosc 7 astfel de straturi), pe care sunt aşezaţi electronii. Fiecare strat electronic este format din mai multe substraturi electronice. Există 4 tipuri de substraturi electronice şi anume `s` – cu 2 orbitali, `p` – cu 3 orbitali, `d` – cu 5 orbitali şi `f` – cu 7 orbitali. Pe fiecare orbital se pot afla cel mult 2 electroni.

În scrierea configuraţiei electronice se precizează, în ordine, substraturile ocupate cu electroni, precum şi numărul de electroni pe care îl conţine fiecare substrat. Numărul stratului din care face parte substratul respectiv precede literei ce reprezintă tipul substratului, iar numărul de electroni din substratul respectiv succede această literă (`2p6` => stratul este 2, numărul de electroni este 6).

Conform principiului energetic, electronii ocupă orbitalii atomici în ordinea creşterii energiei, începând cu orbitalul de cea mai joasă energie. Deci ordinea de completare a orbitalilor cu electroni este dată de direcţia săgeţilor din următoarea schemă:



Se cere:

a) Dându-se numărul atomic al unui atom (este egal cu numărul de electroni ai atomului respectiv) să se afişeze configuraţia sa electronică.

Exemplu. `Z=20` se obţine configuraţia "`1s2 2s2 2p6 3s2 3p6 4s2`" (substraturile se vor despărţi între ele printr-un spaţiu.)

b) Dându-se configuraţia electronică a unui atom să se deducă numărul său atomic precum şi numărul de electroni de pe ultimul strat (care este egal cu valenţa elementului respectiv).

Exemplu. Elementul cu configuraţia "`1s2 2s2 2p6 3s2 3p3`" are numărul atomic `Z=15`, iar valenţa este 5.

54. Fiind dat un careu rebusistic, să se determine toate cuvintele care apar în acel careu şi să se verifice dacă un cuvânt apare de mai multe ori. Pătrăţelele "negre" din careu sunt date utilizând caracterul "*" Se consideră că un cuvânt are cel puţin 2 caractere.

Exemplu. Pentru m=12, n=12 şi careul

	P		O		S		C		I		D
P	R	E	T	U	T	I	N	D	E	N	I
	O	M	E	N	I	E		O	R	A	S
U	N	I	T	I		P	A	S	N	I	C
	O	S		C	H	E	L		I	V	I
O	S		C	A	R		G	S		A	P
	T	R	O	T	I	N	E	T	A		L
B	I	E	T		B	A		E	U	R	I
	C	L	E	R		P	L	A	T	A	N
P	U	I		A	S	I	A		U	T	A
	R	E	G	I	M		S	T	R	A	T
C	I	F	R	E		S	A	R	I	T	I

cuvintele sunt PRETUTINDENI, OMENIE, ORAS, UNITI, PASNIC, OS, CHEL, IVI, CAR, GS, AP, TROTINETA, BIET, BA, EURI, CLER, PLATAN, PUI, ASIA, UTA, REGIM, STRAT, CIFRE, SARITI, PRONOSTICURI, EMIS, RELIEF, OTET, COTE, GR, UNICAT, RAIE, STI, HRIB, SM, IEPE, NAPI, CN, ALGE, LASA, iar cuvântul 'OS' este singurul care se repetă.

4.2. Tipul mulţime (set of)

55. Se citesc n cuvinte. Să se afişeze literele distincte din fiecare cuvânt şi din toate cuvintele.

Exemplu. Dacă cuvintele citite sunt: masa, elefant, mouse mulţimile afişate vor fi: ['a', 'm', 's'], ['a', 'e', 'f', 'l', 'n', 't'], ['e', 'm', 'o', 's', 'u'] şi respectiv ['a', 'e', 'f', 'l', 'm', 'n', 'o', 's', 't', 'u'].

56. Se citeşte de la tastatură un şir de numere întregi cuprinse între 0 şi 255 până la întâlnirea valorii 0. Să se afişeze numerele distincte dintre cele citite.

Exemplu. Pentru şirul de numere 3, 6, 4, 6, 1, 3, 8, 5, 8, 3, 9, 0 se va afişa şirul 1, 3, 4, 5, 6, 8, 9.

57. Scrieți un program care citește de la tastatură mai multe perechi de numere naturale, până ce se introduce perechea `(0,0)` și creează două mulțimi: prima mulțime va memora șirul primelor valori din fiecare pereche, iar a doua mulțime va conține șirul valorilor secundare din fiecare pereche. Elementele perechii `(0,0)` nu se vor introduce în mulțimi. Programul va afișa cele două mulțimi create precum și intersecția acestora.

Exemplu. Dacă de la tastatură se introduc perechile `(8,2)`, `(9,0)`, `(4,8)`, `(0,0)`, mulțimile vor fi `{8,9,4}`, `{2,0,8}` și `{8}`.

58. Se citesc de la tastatură două mulțimi `A` și `B`, cu câte `m` respectiv `n` elemente numere întregi cuprinse între `1` și `255`. Elementele unei mulțimi sunt citite toate de pe aceeași linie, separate prin spații. Să se tipărească mesajul '`da`' dacă mulțimile sunt egale, respectiv '`nu`' în caz contrar.

Exemplu. Dacă se introduce `m=5`, `n=8` și mulțimile `{1 2 3 4 5}`, `{2 3 4 1 5 2 5 3}` atunci se va afișa mesajul '`DA`'

59. Pentru `x`, `y` și `z` declarate `set of 2..100`, să se obțină mulțimile:
`x` = mulțimea tuturor numerelor $\in$ [2,100]
`y` = mulțimea numerelor prime $\in$ [2,100]
`z = x-y`.

60. Construiți un algoritm eficient care determină toate perechile de numere naturale `a`, `b`, cu `a≤b`, numerele ce formează o pereche având proprietatea că nu au nici o cifră comună și suma lor este egală cu `S`. Valoarea `S` este un număr natural citit de la tastatură (`S<100.000.000`). Fiecare pereche se va scrie pe un rând al ecranului, cu un spațiu între elementele ce compun perechea.

Exemplu. Pentru `S=16`, se vor afișa (nu neapărat în această ordine) perechile: `(2,14)`, `(6,10)`, `(4,12)`, `(5,11)`, `(0,16)`, `(7,9)`.

61. Să se scrie un program care afișează valorile comune tuturor liniilor unei matrice `A` cu `m` linii și `n` coloane cu componente de tip `byte`.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 5 & 6 & 9 & 1 & 10 & 2 \\ 2 & 1 & 13 & 23 & 94 & 17 \\ 12 & 2 & 32 & 10 & 112 & 1 \end{pmatrix}$ se va afișa 1 și 2.

62. Să se genereze mulțimile

$$A = \{x;\ \ x \le n,\ 7x + 1 \text{ se divide prin } 5\}$$

$$B = \{x;\ \ x \le n,\ x^2 + 1 \text{ se divide prin } 3$$

unde numărul întreg `n` (`n<=255`) se citește de la tastatură. Să se afișeze elementele celor două mulțimi.

Exemplu. Pentru `n=20` se obțin mulțimile `A={2, 7, 12, 17}` și `B={}`.

63. Să se afişeze cel mai mare număr obţinut din cifrele distincte ale unui număr întreg dat.

Exemplu. Pentru numărul `7252382` se va afişa `87532`.

64. Să se determine cifrele comune a două numere.

Exemplu. Cifrele comune ale numerelor `235142` şi `38721` sunt `1, 2, 3`.

65. Se citesc `n` numere naturale de cel mult `9` cifre fiecare. Să se afişeze cifrele comune tuturor numerelor.

Exemplu. Pentru `n=4` şi numerele `12345, 51287, 23391, 2318` cifrele comune sunt `1` şi `2`.

66. Se citesc numele medicamentelor indicate şi a celor contraindicate în `n` boli. Se cere să se afişeze numele medicamentelor care se pot administra unor pacienţi care suferă de mai multe boli din cele `n`.

Exemplu. Dacă `n=3` iar bolile sunt `'Anemia'`, `'Gripa'`, `'Ulcer'`, medicamentele indicate în aceste boli sunt (`'Vitamina A'`, `'Vitamina B'`, `'Vitamina C'`, `'Fier'`), (`'Vitamina C'`, `'Aspirina'`, `'Paracetamol'`) şi respectiv (`'Ulcerostop'`, `'Vitamina B'`) iar medicamentele contraindicate sunt `( )`, `('Penicilina')`, `('Aspirina')` atunci unui pacient care suferă de `Ulcer` şi `Gripă` i se pot administra următoarele medicamente: `'Vitamina C'`, `'Paracetamol'`, `'Ulcerostop'`, `'Vitamina B'`.

67. Se citesc `n` mulţimi de numere naturale mai mici sau egale cu `255`. Se cere să se precizeze dacă aceste mulţimi sunt disjuncte două câte două sau nu. (Două mulţimi sunt disjuncte dacă nu au elemente comune).

Exemplu. Dacă `n=4` mulţimile `{1,2,3}`, `{4}`, `{5,6}`, `{7,8,9,10}` sunt disjuncte două câte două.

68. Se dă o mulţime de culori precum şi combinaţiile ce formează drapelele a `n` ţări. Să se determine mulţimea culorilor comune pentru cel puţin trei drapele.

Exemplu. Dacă avem `7` drapele cu următoarele culori `(roşu,galben, albastru)`, `(verde,alb,negru)`, `(roz,violet,galben)`, `(violet, galben,roşu)`, `(roşu,alb,albastru)`, `(alb,albastru,negru)`, `(galben,gri,maro)` atunci culorile care apar pe cel puţin `3` drapele sunt: `alb`, `albastru`, `galben`, `roşu`.

69. Mulţimile `A` şi `B` sunt părţi ale mulţimii `{1,2,…,250}`. Pentru fiecare mulţime se introduc de la tastatură elementele sale, lista elementelor considerându-se încheiată când a fost tastată o valoare negativă. Să se afişeze mulţimile `A`, `B`, `(A∪B)\(A∩B)` şi `(A\B) ∪(B\A)` sub forma cunoscută din matematică.

Exemplu. Dacă `A={1,2,4,6,7,9}` şi `B={2,3,5,6,8,9,10,11}` atunci `(A∪B)\(A∩B)= (A\B) ∪(B\A)={1,3,5,7,8,10,11}`.

70. Se citeşte un şir de n mulţimi de numere de tip byte. Se cere să se afişeze elementele complementarei intersecţiei mulţimilor faţă de reuniunea lor.

Exemplu. Dacă mulţimile sunt {1,2,3,4,5,20}, {2,3,8,20,41}, {1,2,4,12,20} atunci complementara intersecţiei faţă de reuniune este {1,3,4,5,8,12,41}.

71. Pentru un text dat să se determine mulţimea cifrelor codurilor ASCII folosite pentru reprezentarea textului.

Exemplu. Dacă textul este 'Exemplu' atunci mulţimea cifrelor codurilor ASCII este {0,1,2,6,7,8,9}.

72. Pentru un text dat să se afişeze mulţimea vocalelor, a consoanelor, a cifrelor şi a caracterelor speciale folosite în acel text. Se va folosi tipul mulţime.

Exemplu. Pentru textul "Exemplu de NOTA ** 10 **." se vor afişa următoarele mulţimi: vocale {'e','u','A','E','O'}, consoane {'d','l','m','p','x','N','T'}, cifre {'0','1'}, caractere speciale {' ','*','.'}.

73. Se citesc de la tastatură n numere naturale cuprinse între 0 şi 9999. Realizaţi un program care stabileşte dacă există cifre zecimale care nu apar în componenţa nici unuia dintre cele n numere. Dacă există astfel de cifre acestea se vor afişa, iar în caz contrar se va tipări un mesaj.

Exemplu. Pentru n=5 şi numerele 124, 281, 632, 10, 144 se vor afişa cifrele 5,7,9.

74. Se citesc de la tastatură pe rând n cuvinte formate din litere mici ale alfabetului englez. Să se afişeze perechile de cuvinte din cele citite cu proprietatea că cele două cuvinte din pereche folosesc aceeaşi mulţime de litere distincte.

Exemplu. Cuvintele acasa şi sac conţin litere din aceeaşi mulţime {'a','c','s'}.

75. Se citesc de la tastatură informaţii sub forma a n perechi de elevi (i,j) (date prin câte două litere mici), având semnificaţia că elevul i este în clasa cu elevul j. Se cere să se precizeze care sunt toate clasele şi ce elevi sunt în fiecare clasă.

Exemplu. Pentru n=10 şi perechile ('a','b'), ('a','d'), ('c','e'), ('f','u'), ('p','e'), ('a','p'), ('i','t'), ('k','l'), ('t','u'), ('s','l'), se obţin 3 clase cu următoarele componenţe: ('a','b','c','d','e','p'), ('f','i','t','u'), ('k','l','s').

76. Să se determine toate numerele de trei cifre care au proprietatea că atât ele cât şi pătratul lor sunt scrise cu cifre distincte, diferite pentru număr şi pătratul lui.

Exemplu. Primul număr de forma cerută este 209 (209^2=43681).

77. Să se determine toate numerele mai mici ca 10000 cu proprietatea că pătratul numărului este format din aceleaşi cifre ca şi numărul.

Exemplu. Un astfel de număr este 4762 (4762^2=22676644).

78. Se dau numerele naturale a şi b mai mici decât 255. Să se determine divizorii comuni ai celor două numere date, folosind tipul mulţime.

Exemplu. Pentru a=234 şi b=126 se vor afişa numerele 1, 2, 3, 6, 9, 18.

79. Să se determine numerele prime mai mici ca n, n<256, folosind "Ciurul lui Eratostene" simulat cu ajutorul tipului mulţime.

Exemplu. Pentru n=20 se vor afişa numerele 2, 3, 5, 7, 9, 11, 13, 17, 19.

4.3. Tipul înregistrare (record)

80. Tocmai s-a încheiat primul tur de scrutin pentru disputatele alegeri ale primarului din oraşul dumneavoastră. Să facem o analiză a voturilor primite de cei patru candidaţi la scaunul de primar. Dorim să aflăm câte voturi a primit fiecare dintre candidaţi în fiecare cartier al oraşului şi câte voturi totale a primit fiecare candidat în parte. Imaginaţi o structură de date adecvată rezolvării acestei probleme şi scrieţi apoi un program care o rezolvă.

81. O companie vrea să ştie procentul salariilor totale şi cheltuielile totale atribuibile fiecărui angajat. Pentru fiecare angajat se introduce de la tastatură o linie de date conţinând: numele persoanei (maxim 20 de caractere), urmat de o virgulă, urmată de salariul total al persoanei (întreg) şi cheltuielile (real).

Scrieţi un program care realizează un raport cu un header conţinând totalele salariilor şi cheltuielile totale pe firmă. După acest header să se afişeze un tabel conţinând câte o linie pentru fiecare angajat astfel: numele, prenumele, procentul salariului total, procentul cheltuielilor totale. Procentul se calculează relativ la totalul pe firmă.

82. Se citesc data naşterii unei persoane şi data curentă. Să se stabilească vârsta în ani.

Exemplu. Dacă data naşterii este 30.10.1980 iar data curentă este 01.06.2002 atunci persoana are 21 de ani.

83. Se citesc două fracţii p şi q, sub forma (numărător, numitor). Să se afle forma ireductibilă a fracţiei p+q.

Exemplu. $\frac{2}{9} + \frac{2}{3} = \frac{8}{9}$.

84. Se citesc `n` numere complexe sub forma (`parte reală, parte imaginară`). Să se afişeze ordonat crescător şirul modulelor acestor numere complexe.

Exemplu. Pentru n=5 şi numerele `2+3i, 3-i, 7-2i, 1-i, 3i`, se va afişa `1.4142, 1.7320, 3.1622, 3.6055, 7.2801`.

85. La un concurs de informatică au participat `n` elevi, pentru fiecare elev cunoscându-se numele, liceul de proveniență (şiruri de caractere) şi nota obținută (de la `1` la `10`). Realizați un program care citeşte datele elevilor participanți, apoi tipăreşte numele elevilor cu cea mai mare notă, precum şi media generală a concurenților.

Exemplu. Pentru `n=4` şi următoarele date: (`Ionescu,CNGL,10`), (`Antonescu,Goga,8`), (`Iliescu,CNGL,9`), (`Lazaroiu,LIA,10`), elevii cu cea mai mare notă sunt `Ionescu` şi `Lazaroiu` iar media generală a tuturor elevilor este `9.25`.

86. Definiți structurile de date necesare pentru evidența cărților existente într-o bibliotecă, ştiind că pentru fiecare carte se înregistrează: titlul cărții, prețul, numărul de exemplare şi valoarea cărții respective (prețul * numărul de exemplare). Scrieți apoi un program care, prin intermediu unui meniu, selectează în mod repetat, atâta timp cât utilizatorul doreşte acest lucru, una din următoarele acțiuni posibile: introducerea datelor aferente unei noi cărți, determinarea valorii totale a cărților existente în bibliotecă şi afişarea informațiilor aferente tuturor cărților.

87. Scrieți un program care să citească temperaturile măsurate din oră în oră precum şi cantitățile zilnice de precipitații dintr-o lună a anului. Apoi, programul trebuie să afişeze:

    a) temperatura maxime (împreună cu ziua şi ora asociată)

    b) temperatura minimă (împreună cu ziua şi ora asociată)

    c) lista zilelor, ordonată descrescător în funcție de cantitatea de precipitații

    d) media precipitațiilor zilnice din luna respectivă a anului.

88. Scrieți un program pentru evidența studenților unei facultăți, care să permită alegerea repetată a uneia din opțiunile de mai jos, cât timp utilizatorul doreşte aceasta:

    adăugarea unui student în grupă;

    listarea tuturor studenților din grupă;

    afişarea informațiilor despre un anumit student, căutat după nume.

    Despre fiecare student al unei grupe se cunosc numele studentului, media la sfârşitul unei sesiuni, şi valoarea bursei.

89. Se citesc de la tastatură date despre situaţia şcolară a `n` studenţi: cod student, grupa, nume student, sex, număr examene (`m`), `nota1`, `nota2`, ..., `notam`. Să se listeze studenţii care au examene nepromovate şi, respectiv, numărul examenelor nepromovate. Un examen este nepromovat dacă nota este sub `5.00`.

90. Se consideră un tablou cu înregistrări ce conţin informaţii referitoare la elevii unei şcoli, organizate astfel: nume, clasa, număr note (`n`), `nota1`, `nota2`, `notan`.

    Scrieţi un program care să ordoneze elevii descrescător după medii apoi stabiliţi clasa cu cei mai mulţi elevi care au toate notele cel puţin `7`.

91. Să se descompune un număr în factori primi, memorând rezultatul sub forma unui vector de înregistrări; fiecare înregistrare va cuprinde două câmpuri, unul indicând factorul prim, iar celălalt puterea la care apare în descompunere.

    Exemplu. Pentru `n=1960` se obţine vectorul `((2,3), (5,1), (7,2))` unde primul număr din fiecare paranteză reprezintă factorul prim iar al doilea reprezintă exponentul.

92. Să se scrie un program care să administreze un parc de automobile. Informaţiile relative la un automobil sunt: numărul de locuri, puterea (în cai putere), marca, numărul de înmatriculare, tipul de carburant (benzină sau motorină), natura (berlină, break sau decapotabilă). Programul trebuie să permită intrarea unei maşini, ieşirea unei maşini, înlocuirea unei maşini cu alta de acelaşi model (având alt număr de înmatriculare).

93. Se consideră un vector `a` cuprinzând articole de tipul:

    ```
    record  x,y: integer;  end;
    ```

    a) Să se ordoneze vectorul `a` descrescător după câmpul `x`, iar în caz de egalitate se sortează crescător după câmpul `y`;

    b) Să se ordoneze acele componente ale lui `a` care au câmpul `x` impar, crescător după câmpul `y`, celelalte componente ale lui `a` rămânând pe poziţia lor (cea de după punctul a)).

    c) Să se calculeze suma tuturor câmpurilor `x` şi produsul tuturor câmpurilor `y` pare din `a`.

    Exemplu. Pentru `n=10` şi `a=((1,5), (2,1), (7,7), (9,5), (2,6), (12,2), (4,4), (-1,4), (3,8), (6,22))` se afişează:

    a) `(12,2), (9,5), (7,7), (6,22), (4,4), (3,8), (2,1), (2,6), (1,5), (-1,4);`

    b) `(12,2), (-1,4), (9,5), (6,22), (4,4), (1,5), (2,1), (2,6), (7,7), (3,8);`

    c) `s=45, p=5913600.`

94. Să se scrie un program care determină zodia pentru o dată de naştere furnizată de la tastatură, sub forma unui număr întreg reprezentând ziua naşterii şi un şir de caractere pentru luna de naştere. Se va folosi o structură de forma:

```
zodii: array [1..12] of
            record
                  zi_sf   1..31;
                  luna    string;
                  nume_zodie   string;
            end;
```

unde `zi_sf`, `luna` reprezintă ultima dată din zodia respectivă.

Observaţie. Amintim zodiile şi perioadele corespunzătoare:

Capricorn	23 decembrie - 19 ianuarie
Vărsător	20 ianuarie - 19 februarie
Peşti	20 februarie - 20 martie
Berbec	21 martie - 19 aprilie
Taur	20 aprilie - 20 mai
Gemeni	21 mai - 20 iunie
Rac	21 iunie - 21 iulie
Leu	22 iulie - 22 august
Fecioară	23 august - 22 septembrie
Balanţă	23 septembrie - 22 octombrie
Scorpion	23 octombrie - 21 noiembrie
Săgetător	22 noiembrie - 22 decembrie.

Exemplu. Data de 5 septembrie este în zodia Fecioarei.

95. Se dau `n` puncte în plan prin coordonatele lor `(x,y)`. Să se listeze toate punctele care se află în interiorul cercului `C(`x_0`,`y_0`,R)`.

Exemplu. Dacă `n=10` şi punctele sunt `(1,5)`, `(2,1)`, `(7,7)`, `(9,5)`, `(2,6)`, `(12,2)`, `(4,4)`, `(-1,4)`, `(3,8)`, `(6,22)`, iar x_0=2, y_0=1, `R=5`, se vor afişa punctele `(1,5)`, `(2,1)`, `(4,4)`, `(-1,4)` (se observă că punctul `(2,6)` se află pe conturul cercului deci nu face parte din interiorul acestuia).

96. Se dau `n` puncte în plan, prin coordonatele lor `(x,y)`. Să se afişeze indicii punctelor aflate pe prima bisectoare şi indicii punctelor aflate pe una din axele `Ox` sau `Oy`.

Exemplu. Pentru `n=7` şi punctele `(3,7)`, `(0,20)`, `(3,3)`, `(5,0)`, `(0,10)`, `(-5,-5)`, `(4,9)` punctele 3 şi 6 se află pe prima bisectoare iar punctele 2, 4 şi 5 se află pe axe.

97. Scrieţi un program care simulează amestecarea unui pachet de cărţi de joc (cu 52 de cărţi).

98. Să se scrie un program pentru admiterea la un liceu care să permită:

a) inițializarea vectorului de înregistrări de tip elev;

b) adăugarea unui elev;

c) eliminarea unui elev cu un nume dat;

d) eliminarea elevului de pe o poziție dată;

e) calcularea mediilor;

f) listarea alfabetică a elevilor;

g) listarea elevilor în ordine descrescătoare a mediilor;

h) listarea tuturor elevilor cu medii peste o medie dată.

Se presupune că examenul constă din două probe.

99. Se citesc de la tastatură date despre materialele aflate într-o magazie:

codul materialului (întreg)
denumirea materialului
preț
stoc normat
cantitate intrată
cantitate ieșită.

Să se afișeze pe ecran (sub formă de tabel) o situație a materialelor cu stoc supranormativ (cantitate intrată - cantitate ieșită > stoc normat).

100. Se citesc de la tastatură date despre angajații unei firme. Să se afișeze pe ecran (sub formă de tabel) angajații care îndeplinesc concomitent condițiile de funcție și sex, egale cu valorile introduse de la tastatură. Datele despre angajați sunt următoarele:

marca
numele și prenumele
codul funcției
sex
salariu

Corespondența dintre codul funcției și denumirea funcției este memorată într-un vector de articole.

La afișare va apărea denumirea funcției în locul codului.

101. Să se simuleze funcționarea unui joc LOTO. La joc participă `n` persoane, fiecare cu câte un bilet, pe fiecare bilet se trec `m` numere, iar din urnă se extrag `p` numere cuprinse între `1` și `99`. (`m`, `n`, `p` și numerele câștigătoare se introduc de la tastatură). Pentru fiecare jucător se cunosc: numele, seria biletului cumpărat (două litere și șase cifre), precum și cele `m` numere pe care le-a trecut pe biletul său. Să se afișeze participanți în ordinea descrescătoare a numărului de numere "ghicite". La același punctaj are prioritate cel care a ghicit numere situate mai aproape de începutul extragerii.

102. Despre cei `n` elevi ai unei clase se cunosc numele, înălțimea și greutatea, ultimele două informații sunt numere naturale.

Se cere să se creeze o listă a elevilor, ordonată descrescător în funcție de înălțime, iar la înălțimi egale în funcție de greutate. În final se cere să se calculeze, sub formă de fracție ireductibilă, media rapoartelor înălțime/greutate.

Datele se citesc de la tastatură.

103. La o stație meteo se alcătuiește zilnic, un buletin meteo care conține:
data
numele meteorologului de serviciu;
presiunea atmosferică;
temperatura atmosferică;

din ziua respectivă. Să se afișeze temperatura maximă atinsă în perioada urmărită, precum și zilele în care s-a înregistrat maxima, împreună cu numele meteorologilor de serviciu din zilele respective.

104. Pentru evidența medicamentelor dintr-un depozit, sunt necesare următoarele informații:

codul medicamentului (întreg);

denumire (șir de 30 de caractere);

prețul medicamentului (întreg);

Presupunând că în depozit există `n` medicamente se cere să se majoreze prețurile tuturor medicamentelor cu `p%` și să se afișeze medicamentele cu noile prețuri,

$$p = \begin{cases} 5 & \texttt{daca codul medicamentului} \leq 1000 \\ 10 & \texttt{in caz contrar} \end{cases}$$

4.4. Tipul fișier

105. Să se afișeze numărul de numere de pe fiecare linie a unui fișier text cu numele citit de la tastatură.

Exemplu. Dacă numele fișierului citit este `F1.TXT` iar conținutul acestui fișier este:

```
12 4 67 43 878
2 7 8 5 3 2 89 123
9
```

valorile afișate vor fi `5, 8, 1.`

106. Se scrie un program care afişează pe ecran conţinutul unui fişier text cu nume citit de la tastatură.

107. Să se afişeze numărul de cuvinte de pe fiecare linie a unui fişier text cu numele citit de la tastatură. Cuvintele se consideră separate între ele prin unul sau mai multe spaţii.

108. Fişierul 'C1.TXT' conţine una sau mai multe linii cu cuvinte (conţinând litere mari sau mici ale alfabetului englez) separate între ele prin unul sau mai multe spaţii. Să se copieze aceste cuvinte în fişierul 'C2.TXT' câte k cuvinte pe o linie, despărţite între ele printr-un singur spaţiu.

Exemplu. Pentru k=3 şi conţinutul fişierului C1.TXT următorul

```
exemplu    de  program
simplu         dar      nu banal  este      aici    timpul
sa
ne      jucam  cu fisierele text
```

atunci conţinutul fişierului C2.TXT va fi:

```
exemplu de program
simplu dar nu
banal este aici
timpul sa ne
jucam cu fisierele
text
```

109. Din fişierul text 'FISIN.TXT' se citesc mai multe numere întregi, scrise pe unul sau mai multe rânduri. Să se copieze în fişierul 'FISOUT.TXT' acele numere din fişierul de intrare care au cel mult două cifre, toate pe primul rând al fişierului de ieşire separate prin spaţiu, şi să se afişeze pe ecran câte astfel de numere s-au găsit.

Exemplu. Dacă fişierul FISIN.TXT are următorul conţinut:

```
12 5 784 2374
23 5 6657 34 222
4 890
```

fişierul FISOUT.TXT va fi

```
12 5 23 5 34 4
```

iar pe ecran se va afişa valoarea 6.

110. Să se creeze un fişier text care să conţină toate numerele cuprinse între a şi b formate din cifre identice. Valorile lui a şi b se citesc de la tastatură.

Exemplu. Pentru a=10 şi b=150 fişierul de ieşire va arăta astfel:

```
11 22 33 44 55 66 77 88 99 111
```

111. În fişierul text 'PROBL.TXT' se găsesc pe prima linie două numere a şi b iar pe următoarea linie mai multe numere întregi. Să se verifice dacă toate aceste numere se găsesc în intervalul [a,b] sau se găsesc toate în afara acestui interval. Se va afişa pe ecran un mesaj corespunzător.

Exemplu. Dacă conţinutul fişierului PROBL.TXT este următorul:

```
20 25
2 6 8 30 55 3 -22 4 103
```

se va afişa mesajul "DA" (sau unul cu acelaşi sens).

112. Să se copieze conţinutul unui fişier text în alt fişier, înlocuind literele mici cu litere mari, cu excepţia şirurilor scrise între ghilimele, care rămân nemodificate.

Exemplu. Dacă fişierul de intrare este

```
fvbjHsdfkj GJHLvhvhHG "bjhHJv"  sdhBv "abT" uv
```

atunci fişierul de ieşire este

```
FVBJHSDFKJ GJHLVHVHHG "bjhHJv"  SDHBV "abT" UV
```

113. Se dau două şiruri de caractere S şi T şi un fişier text IN.TXT. Să se scrie un program care copiază conţinutul lui IN.TXT într-un alt fişier text OUT.TXT, însă înlocuieşte fiecare apariţie a şirului S cu şirul T.

Exemplu. S='abc', T='12345'

IN.TXT	OUT.TXT
uvabcxyabcmn	uv12345xy12345mn
cabcde	c12345de

114. Se citesc din fişierul text 'ABC.TXT' un şir de numere întregi aflate toate pe prima linie a fişierului, separate între ele prin spaţii. Se cere să se determine câte din numerele citite sunt mai mici decât media aritmetică a tuturor numerelor. Rezultatul se va afişa pe ecran.

Exemplu. Dacă ABC.TXT este:

```
45 678 23 12 5 89 4 21
```

atunci se va afişa valoarea 7 (media este 109.625 deci în afară de-numărul 678 toate sunt mai mici decât această media)

115. Să se scrie un program care şterge fiecare apariţie a unui şir S dintr-un fişier text cu numele citit de la tastatură.

Exemplu. Dacă iniţial fişierul era:

```
abc fdg dgh ergadfv abc sdf
esrgsfb erg abc fgsdfb fgabcde szdg
```

iar s="abc" atunci în final fişierul va arăta astfel:

```
 fdg dgh ergadfv  sdf
esrgsfb erg  fgsdfb fgde szdg
```

116. Pe prima linie a fişierului text 'LITERE.TXT' se găseşte o succesiune de cel puţin două şi cel mult 2000 de caractere, caractere ce pot fi doar litere şi spaţii. Scrieţi un program care citeşte de la tastatură un număr natural k (1<k<1000) şi stabileşte dacă există în fişier vreo literă care apare de exact k ori. Programul afişează pe ecran mesajul "DA" urmat de literele care îndeplinesc condiţia amintită, sau "NU" dacă nu exista astfel de litere.

Exemplu. Dacă fişierul LITERE.TXT are următorul conţinut:

```
zdfb sdfgn hfmfgvbasg asefnetyhj sdfgbasdfgsrtn
```

iar k=3 atunci se vor afişa mesajul "DA" şi literele b, n, a.

117. Se dă un fişier text cu numele IN1.TXT, care conţine codurile ASCII ale unor caractere, separate prin spaţiu. Să se construiască fişierul OUT1.TXT care să conţină caracterele cu codurile ASCII din fişierul IN1.IN.

Exemplu.

IN1.TXT	OUT1.TXT
103 42 116 87 33	g * W !
91 76	[L
110 92 124	n \ |

118. Se citeşte un şir de numere întregi din fişierul 'numere.txt'. Fişierul conţine: pe primul rând numărul n de elemente ale şirului, apoi, pe fiecare din următoarele n rânduri, câte un element. Să se afişeze suma elementelor pozitive ale şirului.

Exemplu.

numere.txt

```
5
6
7
-2
4
-10
```

se va afişa suma 17.

119. Se citeşte un şir de numere întregi din fişierul 'numere2.txt'. Fişierul conţine: pe primul rând numărul n de elemente ale şirului, apoi pe al doilea rând elementele şirului separate prin spaţii. Să se afişeze elementele pare din şir.

Exemplu.

numere.txt

```
5
6 7 -2 4 -11
```

se vor afişa valorile 6 -2 4.

120. Să se memoreze în fișierul 'MATR.TXT' primele n (n citit de la tastatură) linii din tabloul bidimensional (având formă triunghiulară) A ale cărui elemente se obțin astfel:

a) A(i,1)=A(i,i)=1 pentru i>=1;

b) A(i,j)=A(i-1,j-1)+A(i-1,j) pentru orice i>=3 și 2<=j<=i-1.

c) A(i,j)=A(j,i) pentru i<>j.

Exemplu. Pentru n=4 conținutul fișierului MATR.TXT va fi:

```
1 1 1 1
1 1 2 3
1 2 1 3
1 3 3 1
```

121. Din fișierul 'NR.IN' se citesc mai multe numere întregi. Ele sunt scrise în fișier pe mai multe linii, două numere de pe aceeași linie fiind separate prin spațiu. Să se scrie un program care construiește fișierul text 'NR.OUT' conținând numărul cel mai mare și numărul cel mai mic din fiecare linie a fișierului 'NR.IN' Cele două numere se vor scrie pe aceeași linie separate prin spațiu.

Exemplu.

NR.IN	NR.OUT
12 342 56 3 21	3 342
1 21 -7 5	-7 21
3	3 3
90 1 902	1 902
2 2 2	2 2

122. Scrieți un program care citește din fișierul text 'IN.TXT' mai multe numere naturale de cel mult trei cifre fiecare și afișează pe ecran câte numere pare și câte numere impare există în fișier. Numerele sunt scrise pe prima linie a fișierului, cu câte un spațiu între ele.

Exemplu. Dacă în fișierul IN.TXT se găsesc următoarele numere:

```
12 65 352 346 763
```

atunci se va afișa: 3 numere pare și 2 numere impare.

123. Să se afișeze litera (literele) care se întâlnește cel mai des într-un fișier text ce conține litere mari și mici ale alfabetului englez, spațiu și cifre zecimale.

Exemplu. Pentru fișierul:

```
zdfb sdfgn hfmfgvbasg
asefnetyhj sdfgbasdfgsrtn
```

se va afișa litera f care apare de 7 ori.

124. Fiind dat un fişier care conţine mai multe linii de text de lungime variabilă, scrieţi un program care afişează linia (liniile) de lungime maximă.

Exemplu. Fişierul de intrare este:

```
zsdfvdfb dfb asdfg erg
dfbdb dffgasgnjm dcvdf fbsdfb
nfgn
dsfgnsdfg sdfgnnyk fghn34tsdr
```

se afişează pe ecran:

```
dfbdb dffgasgnjm dcvdf fbsdfb
dsfgnsdfg sdfgnnyk fghn34tsdr
```

125. Fiind dat un fişier care conţine mai multe linii de text de lungime variabilă, scrieţi un program care copiază în alt fişier text linia (liniile) de lungime maximă.

126. Se dau mai multe fişiere text astfel: primul fişier numit "**BAZA.IN**" conţine pe fiecare linie numele câte unui alt fişier text, fiecare din aceste fişiere existând pe hardisk şi conţinând mai multe linii de text fiecare. Un alt fişier text numit "**SECVENTA.TXT**" conţine, pe o linie un şir de caractere. Problema este de a căuta şirul din fişierul "**SECVENTA.TXT**" în toate fişierele text a căror nume se află memorate în "**BAZA.IN**" Pentru fiecare întâlnire a şirului respectiv în unul din fişiere se va memora în fişierul "**BAZA.OUT**" câte o linie compusă din: numele fişierului în care s-a întâlnit şirul, linia în care s-a întâlnit şi poziţia din care începe şirul în linia respectivă.

Exemplu.

BAZA.IN	**SECVENTA.TXT**	**F1.TXT**	**F2.TXT**
f1.txt	abc	xyabcmnabc	xyz
f2.txt		abxy	
f2.txt		abdabcabo	

F3.TXT	**BAZA.OUT**
abcabcabc	F1.TXT 1 3
nmabc	F1.TXT 1 8
	F1.TXT 3 4
	F3.TXT 1 1
	F3.TXT 1 4
	F3.TXT 1 7
	F3.TXT 2 3

127. Realizaţi un program care concatenează două fişiere text ("lipeşte" conţinutul celui de-al doilea la sfârşitul primului).

Exemplu. Primul fişier	Al doilea fişier	Fişierul final
abcs	**abgje**	**abcs**
	csdvzsdf	**abgje**
		csdvzsdf

128. Se dă un fişier text conţinând mai multe linii. Să se copieze aceste linii în alt fişier text în ordinea descrescătoare a lungimilor acestor linii.

Exemplu. Pentru fişierul de intrare:

```
zsdfvdfb dfb asdfg erg
dfbdb dffgasgnjm dcvdf fbsdfb
nfgn
dsfgnsdfg sdfgnnyk fghn34tsdr
```

fişierul de ieşire va fi:

```
dfbdb dffgasgnjm dcvdf fbsdfb
dsfgnsdfg sdfgnnyk fghn34tsdr
zsdfvdfb dfb asdfg erg
nfgn
```

129. Se dă un fişier text cu mai multe linii de numere întregi, separate între ele printr-un spaţiu. Să se determine câte din aceste numere sunt palindroame (numărul este egal cu oglinditul său). Rezultatul se va afişa pe ecran.

Exemplu. Dacă fişierul iniţial este:

```
1221 45 23
234 767 452 93
2456
12321
12
```

se va afişa valoarea 3.

130. Se dă un fişier care conţine mai multe linii de text. Să se tipărească, pentru fiecare linie în parte, mulţimea caracterelor distincte din linia respectivă. Rezultatele se vor afişa pe ecran.

Exemplu. Dacă fişierul de intrare este

```
abdbsbg
dsfase
```

se va afişa pe ecran

```
abdgs
adefs
```

131. Să se scrie într-un fişier text următoarea piramidă:

```
A
A A
A A A
```

Numărul de caractere de pe ultima linie se va citi de la tastatură.

132. Se dă un fişier text cu numere întregi, câte unul pe fiecare linie. Să se scrie o variantă a metodei bulelor pentru a sorta conţinutul acestui fişier. Nu se vor folosi tipuri structurate de date.

133. Se dă un fişier text cu numere întregi, câte unul pe fiecare linie. Să se scrie o variantă a metodei de sortare prin inserţie directă pentru a sorta conţinutul acestui fişier. Nu se vor folosi tipuri structurate de date.

134. Se citeşte un şir de numere întregi din fişierul text '`sir.in`'. Toate elementele şirului sunt scrise în fişier pe un singur rând, separate prin spaţii. Nu se cunoaşte numărul elementelor. Să se scrie în fişierul '`sir.out`' elementele distincte ale şirului citit, "grupate" pe două rânduri: pe primul rând se vor scrie elementele pare în ordine crescătoare, iar pe al doilea rând elementele impare în ordine crescătoare. Nu se va folosi nici un algoritm de sortare.

Exemplu. **sir.in**

```
34 23 3 34 39 57 3 124 -3 22
```

sir.out

```
22 34 124
-3 3 23 39 57
```

135. Se dă un fişier text '`IN.TXT`' care conţine mai multe linii de text. Să se creeze un nou fişier '`OUT.TXT`' care conţine liniile din fişierul '`IN.TXT`' precedate de numărul de caractere din linia respectivă. Acest număr va fi separat de restul textului din linie printr-un caracter punct.

Exemplu.

IN.TXT	**OUT.TXT**
sdbm sdgasfd	12.sdbm sdgasfd
fadf	4.fadf
sfgnytaer	9.sfgnytaer
fvd	3.fvd

136. Se dă un fişier text '`A1.TXT`' care conţine pe fiecare rând cuvinte separate între ele prin unul sau mai multe spaţii. Un cuvânt este format din caractere mari şi mici ale alfabetului englez. Să se creeze fişierul '`A2.TXT`' care să conţină liniile fişierului '`A1.TXT`' în ordinea descrescătoare a numărului de cuvinte.

Exemplu. Dacă fişierul **A1.TXT** este

```
exemplu de program
simplu dar nu banal este aici timpul
ne jucam cu fisierele text
```

atunci fişierul **A2.TXT** va arăta astfel:

```
simplu dar nu banal este aici timpul
ne jucam cu fisierele text
exemplu de program
```

137. Un fişier text '**IN.TXT**' conţine pe prima linie un număr întreg **n** (**1<=n<=24**). Pe următoarele **n** linii se găsesc câte o literă **c** (mică) a alfabetului englez urmată de un număr întreg **n** despărţite între ele de un spaţiu. Se cere să se construiască un fişier text '**OUT.TXT**' care să conţină pe fiecare linie caracterul **c** repetat de **n** ori.

Exemplu. Pentru fişierul

```
IN.TXT
5
a 3
u 5
c 8
g 4
e 9
```

fişierul **OUT.TXT** va arăta astfel:

```
aaa
uuuuu
cccccccc
gggg
eeeeeeeee
```

138. Se consideră fişierul text '**X1.TXT**' care conţine mai multe linii cu numere întregi separate între ele printr-un spaţiu. Pentru fiecare linie a acestui fişier se va scrie în fişierul '**X2.TXT**' unul din mesajele '**CRESCATOR**', '**DESCRESCATOR**', '**NESORTAT**' după cum numerele din linia respectivă sunt ordonate crescător, descrescător sau nu sunt sortate. Nu se vor folosi tipuri structurate de date.

Exemplu.

X1.TXT	**X2.TXT**
1 5 3 7	NESORTAT
9 5 2	DESCRESCATOR
1 9 4 8	NESORTAT
1 4 7 9 22	CRESCATOR
44 88	CRESCATOR

139. Un fişier text conţine două linii. Prima linie conţine caractere despărţite prin spaţii, iar a două numere întregi despărţite prin spaţii. Să se afişeze da (sau nu) dacă numărul de caractere de pe prima linie este egal cu numărul de numere de pe a doua linie.

Exemplu. Dacă fişierul de intrare conţine următoarele linii:

```
d u y e n g k l
23 56 123 7 4 241 96 345
```

se va afişa **DA**, iar pentru fişierul

```
f j h t r
23 56 123 7 4 241 96 345
```

se va afişa **NU**.

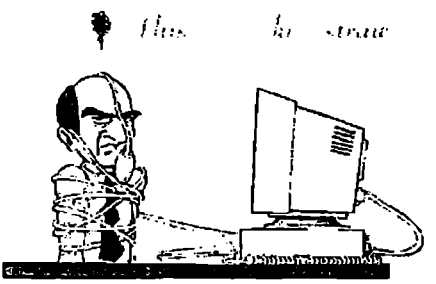
140. Se dă un fișier text cu una sau mai multe linii de numere întregi separate între ele prin spații. Să se creeze un nou fișier text cu numerele din fișierul de intrare, în aceeași ordine, separate între ele printr-un singur spațiu, astfel încât pe un rând să fie cel mult k caractere (cifre sau spații). Pe un rând se vor scrie cât mai multe numere posibil. Valoare k este dată pe prima linie a fișierului de intrare (k apare singur pe linie iar valoarea sa nu se va copia și ea în al doilea fișier).

Exemplu. Pentru fișierul de intrare

```
10
12      56      234  1      346      23
35  647    23  6     87    3     98  45 32
2  45
425 7    34      68    568    3465
```

se obține fișierul

```
12 56 234
1 346 23
35 647 23
6 87 3 98
45 32 2 45
425 7 34
68 568
3465
```



141. Să se scrie într-un fișier text toate numerele prime mai mici decât un număr n dat. Pe un rând se vor scrie p numere prime despărțite printr-un spațiu, excepție ultima linie care poate avea mai puțin de p numere.

Exemplu. Pentru n=20 și p=5 se obține următorul fișier text

```
2 3 5 7 11
13 17 19
```

142. Să se scrie un program care compară, linie cu linie conținutul a două fișiere text, afișând numărul de ordine al celor neidentice.

Exemplu.

```
Primul fișier          Al doilea fișier
sd ghae dsfgzxc        sd ghae dsfg
cfvdfg                 cfvdfg
dfh jjtdfzxc           dfh jjtdfzre
```

se afișează 1, 3.

143. Se consideră o matrice pătrată de ordinul n, cu elemente de tip real, memorată într-un fișier text, fiecare linie a matricei pe o linie a fișierului. Să se scrie un program care determină suma maximelor și suma minimelor de pe coloanele matricei date, fără a aduce întreaga matrice în memorie.

Exemplu. Dacă fișierul de intrare este:

```
23 6 4 9
7 34 2 11
0 3 5 88
1 4 6 2
```

atunci suma maximelor de pe fiecare coloană este `151` iar suma minimelor din fiecare coloană este 7.

144. Fișierul text `V.IN` conține un șir de numere întregi, toate pe un singur rând, separate prin spații. Să se copieze în fișierul text `V.OUT`, unul sub altul, elementele pozitive ale primului fișier.

Exemplu. `V.IN`

```
23 -5 4 9 -8 -10 22 -3 56
```

`V.OUT`

```
23
4
9
22
56
```

145. Se consideră două fișiere text, conținând numere întregi, toate pe aceeași linie a fișierului, sortate crescător. Se cere să se interclaseze cele două fișiere, rezultatul fiind memorat într-un al treilea fișier. Nu se vor folosi tipuri structurate de date.

Exemplu. Dacă primul fișier conține numerele

```
12 16 19 22 30 58
```

iar al doilea

```
1 15 16 20 22 24 35 88 89 102
```

atunci fișierul final va fi

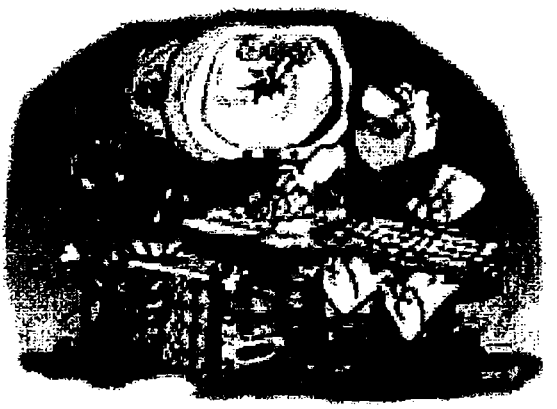
```
1 12 15 16 16 19 20 22 22 24 30 35 58 88 89 102
```

146. Fișierul text '`triunghi.txt`' conține pe fiecare linie câte trei numere separate prin câte un spațiu. Nu se cunoaște câte astfel de triplete sunt în fișier. Pentru fiecare dintre aceste triplete să se verifice dacă pot reprezenta laturile unui triunghi (sunt pozitive și suma a oricare două este mai mare decât al treilea număr): în caz afirmativ se va stabili natura triunghiului: isoscel, echilateral, oarecare. Rezultatele se vor afișa pe ecran.

147. Într-un fişier text se află pe prima linie dimensiunea unei matrice pătrate **A**. Pe următoarele linii se află, câte unul pe linie, elementele matricei **A** în ordinea coloanelor, fără goluri. Matricea nu poate fi stocată în memorie (din cauza dimensiunii foarte mari). Se cere să se calculeze suma elementelor de pe diagonala principală şi suma elementelor de pe diagonala secundară a matricei.

Exemplu. Dacă fişierul este:

```
4
1
2
5
4
87
1
0
23
5
-1
11
4
2
0
18
83
```



acesta reprezintă matricea $\begin{pmatrix} 1 & 87 & 5 & 2 \\ 2 & 1 & -1 & 0 \\ 5 & 0 & 11 & 18 \\ 4 & 23 & 4 & 83 \end{pmatrix}$ deci suma elementelor de pe diagonala principală este **96** iar suma elementelor de pe diagonala secundară este **5**.

148. Fişierul text '**NR1.TXT**' conţine mai multe linii cu numere întregi separate între ele printr-un spaţiu. Să se copieze aceste numere în fişierul '**NR2.TXT**' câte patru numere pe o linie şi astfel încât să se respecte alinierea verticală a numerelor pe coloane. Nu se vor folosi tipuri structurate de date.

Exemplu.

NR1.TXT

```
12  4  67 2341
322 321 35 1 -4 -12 0
1 -3 88
71 102 21 -2 56 32 1 3
```

NR2.TXT

```
 12   4 67 2341
322 321 35    1
 -4 -12  0    1
 -3  88 71  102
 21  -2 56   32
  1   3
```

148. Fişierul text `NR1.TXT` conţine mai multe linii cu numere întregi separate între ele printr-un spaţiu. Să se copieze aceste numere în fişierul `NR2.TXT` câte patru numere pe o linie şi astfel încât să se respecte alinierea verticală a numerelor pe coloane. Nu se vor folosi tipuri structurate de date.

Exemplu. `NR1.TXT`

```
12  4  67 2341
322 321 35 1 -4 -12 0
1 -3 88
71 102 21 -2 56 32 1 3
```

`NR2.TXT`

```
 12   4 67 2341
322 321 35    1
 -4 -12  0    1
 -3  88 71  102
 21  -2 56   32
  1   3
```

149. Se citeşte o matrice din fişierul text `'matrice.in'`. Fişierul conţine:

pe primul rând numărul `m` de linii şi numărul `n` de coloane ale matricei;

pe fiecare din următoarele `m` rânduri, elementele unei linii a matricei, separate prin spaţii.

Să se interschimbe între ele două linii date `L1`, `L2`, scriindu-se matricea rezultată în fişierul `'matrice.out'` (elementele fiecărei linii a matricei pe un rând). Nu se va aduce întreaga matrice în memorie.

Exemplu. `L1=2, L2=5`

`MATRICE.IN`

```
 6 3
 1  2  3
 4  5  6
 7  8  9
10 11
13 14 15
16 17 18
```

`MATRICE.OUT`

```
 1  2  3
13 14 15
 7  8  9
10 11 12
 4  5  6
16 17 18
```

150. Vă propunem următoarea schemă de comprimare-decomprimare a textelor ce nu conţin cifre. Aceasta constă în crearea unei liste de cuvinte din text. La comprimare, orice caracter nealfabetic al textului este copiat direct în textul comprimat. Un cuvânt este copiat în textul arhivat doar dacă este la prima apariţie a lui şi este adăugat la sfârşitul listei de cuvinte. În apariţiile ulterioare cuvântul va fi înlocuit cu numărul de ordine din listă, numerotarea începând de la 1. Cuvintele au cel mult 30 de caractere formate doar din caractere ale alfabetului englez. Scrieţi un program care:

a) Citeşte din fişierul `F1.TXT` un text comprimat, scris pe un o singură linie a fişierului, şi scrie pe ecran textul original.

b) Citeşte din fişierul text `F2.TXT` un text format din mai multe linii şi tipăreşte pe ecran textul arhivat prin metoda descrisă mai sus.

151. Să se rezolve ecuațiile de gradul al doilea ale căror coeficienți se găsesc în fișierul 'coef.txt'. Fișierul conține pe fiecare rând trei numere reale, separate prin spații, reprezentând coeficienții unei ecuații. Pentru fiecare ecuație, se va scrie pe un rând al ecranului: rădăcinile ecuației separate prin spații dacă ecuația este rezolvabilă, respectiv mesajul 'nu are soluții' în caz contrar.

152. Fișierul 'sir1.txt' conține un șir de numere întregi. Elementele șirului se găsesc în fișier toate pe un rând separate prin spații. Să se scrie în fișierul 'sir2.txt', câte unul pe rând, elementele prime ale șirului dat.

Exemplu.

```
sir1.txt
12 4 7 3 90 11 18
```

```
sir2.txt
7
3
11
```

153. Să se scrie programul pentru crearea unui fișier text, cu validarea datelor introduse de la tastatură.

Structura de conversație și de introducere a datelor, este:

```
COD MATERIAL (SAU CTRL/Z): XXX  (Intreg de 3 cifre)
DENUMIRE MATERIAL   XXXX  (şir de lungime 30)
PRET UNITAR   XXXXXXX     (real 7 cifre, 2 zecimale)
STOC NORMAT: XXXXX        (intreg de 5 cifre)
STOC: XXXXX               (intreg de 5 cifre)
```

Datele vor fi supuse următoarelor verificări:

codul materialului: numărul cifrelor să fie egal cu trei;

prețul unitar: numeric mai mare decât 30.50

stoc normat: strict pozitiv;

stoc: pozitiv;

Articolele corecte vor fi memorate, sau nu, într-un fișier text (un material pe o linie), în funcție de răspunsul afirmativ, sau negativ al utilizatorului la întrebarea generată prin program.

În cazul în care un câmp nu este corect se semnalează pe ecran eroarea și se solicită reintroducerea valorii pentru respectivul câmp.

Se va afișa pe ecran numărul de înregistrări transferate în fișier.

154. Pentru evidența angajaților unei firme, se folosește fișierul 'pers.in' care conține: pe primul rând numărul de angajați, apoi, pe fiecare rând, codul, salariul și numele unui angajat, în această ordine, separate prin spații. Realizați un program care scrie în fișierul 'pers.out' datele angajatului (angajaților) cu cel mai mic salariu (fiecare angajat pe câte un rând în același format ca și în fișierul de intrare), apoi afișează pe ecran salariul mediu din firmă.

155. Rezultatele obținute de către elevii unui liceu la Bacalaureat se găsesc în fișierul 'bac.in', care conține: pe primul rând numărul n de elevi, apoi, pe fiecare din următoarele n rânduri media la bacalaureat și numele unui elev, în această ordine, separate printr-un spațiu. Creați fișierul 'bac.out' care conține elevii din fișierul 'bac.in' în ordinea descrescătoare a mediilor (în același format ca și cel din fișierul de intrare), apoi tipăriți pe ecran: numărul elevilor cu media sub 5, numărul elevilor cu media între 5 și 7, și numărul elevilor cu media peste 7.

156. Pentru evidența amenzilor, Direcția Circulație a Poliției folosește fișierul '**amenzi.txt**'. Fiecare amendă aplicată este înregistrată în fișier pe 3 rânduri: numele șoferului contravenient pe primul rând, numărul de înmatriculare al autovehiculului pe al doilea și valoarea amenzii pe al treilea rând. Scrieți un program care tipărește pe monitor, unul sub altul, numele șoferilor a căror amendă este mai mare decât o valoare m dată.

157. Scricți un program care citește de la tastatură două numere naturale n, m ($5<n<m\leq 10^9$) și scrie în fișierul text 'EX.TXT' toate numerele prime din intervalul deschis (n,m). Numerele se scriu în ordine crescătoare, câte 10 numere pe fiecare linie a fișierului, numerele dintr-o linie fiind despărțite între ele prin câte un spațiu.

Exemplu. Dacă n=100 și m=200 atunci fișierul de ieșire va fi:

```
101 103 107 109 113 127 131 137 139 149
151 157 163 167 173 179 181 191 193 197
199
```

158. Se dă fișierul text 'fr.in' care conține pe fiecare linie câte două numere naturale mai mici decât 35000, reprezentând numărătorul respectiv numitorul unei fracții simple.

a) Să se creeze fișierul 'fr.out' care va conține pe fiecare rând numărătorii și numitorii fracțiilor ireductibile rezultate prin simplificarea fracțiilor din fișierul 'fr.in'.

b) Pe ultima linie a fișierului 'fr.out' să se scrie numărătorul și numitorul fracției rezultate prin adunarea fracțiilor din fișierul 'fr.in'. Fracția sumă se va da în formă ireductibilă.

Exemplu.

fr.in	fr.out
2 56	1 28
32 58	16 29
1 5	1 5
3 8	3 8
12 33	4 11
11 2	11 2
	627569 89320

159. Se consideră N cuvinte formate doar din majuscule. Acestea sunt despărțite în silabe cu ajutorul caracterului '-' (minus). Se spune că două cuvinte rimează dacă și numai dacă ultima lor silabă coincide.

Să se scrie un program care împarte cele N cuvinte în grupe, astfel încât oricare două cuvinte aflate în aceeași grupă să rimeze și oricare două cuvinte aflate în grupe diferite să nu rimeze.

După determinarea grupelor, se cere să se caute o frază formată din cât mai multe cuvinte, astfel încât cuvintele simetrice față de mijlocul frazei să rimeze (primul cuvânt va rima cu ultimul, al doilea cu penultimul etc.)

Fișierul de intrare `CUVINTE.IN` conține pe prima linie numărul N al cuvintelor. Pe fiecare dintre următoarele N linii se va afla câte un cuvânt.

Pe prima linie a fișierului de ieșire `CUVINTE.OUT` se va afla numărul K al grupelor formate. Pe fiecare dintre următoarele K linii se vor afla cuvintele dintr-o anumită grupă. Pe ultima linie a fișierului se va afla fraza determinată. Cuvintele de pe o linie vor fi separate printr-un spațiu.

Exemplu.

CUVINTE.IN	CUVINTE.OUT
10	5
MI-SI-U-ME	MISIUNE MINUNE ALOCUTIUNE DIMENSIUNE
MI-NU-NE	LULEA DOILEA
MAR-GEA	CAZ MACAZ
A-LO-CU-TI-U-NE	MARGEA
DI-MEN-SI-U-NE	MISIUNE ALOCUTIUNE LULEA MACAZ MARGEA
DOI-LEA	CAZ DOILEA MINUNE
CAZ	DIMENSIUNE
MA-CAZ	

160. O firmă de telecomunicații are un număr de abonați telefonici ale căror numere de telefon sunt formate din șase cifre, prima fiind obligatoriu diferită de zero. Firma a organizat un concurs. Premiul va fi oferit abonatului al cărui număr de telefon conține doar cifre pare, iar dintre acestea este maxim conform ordinii numerelor naturale. Datele de intrare se citesc din fișierul '`abonați.txt`', care conține pe fiecare rând numele și numărul de telefon al unui abonat, separate prin exact un spațiu.

Să se găsească și să se afișeze pe ecran numele și numărul abonatul câștigător. Dacă nu există un astfel de abonat, se va tipări un mesaj.

161. Fișierul '`text.in`' conține un text, pe fiecare rând fiind scrisă o frază. Cuvintele unei fraze pot fi separate între ele prin caracterele "spațiu", "virgulă" și "punct și virgulă". Să se tipărească în fișierul '`cuvinte.out`', cuvintele distincte din text și frecvențele lor de apariție în cadrul textului (pe fiecare rând se va scrie un cuvânt împreună cu frecvența sa, separate printr-un spațiu).

Exemplu. Pentru fişierul având următorul conţinut:

```
aici, la noi la Sibiu, este mai frig decât la Bucureşti
asta se întâmplă, bineînţeles pentru că suntem la munte
este plăcut şi vara aici
```

se va afişa:

aici 2	decât 1	că 1
la 3	Bucureşti 1	suntem 1
noi 1	asta 1	munte 1
Sibiu 1	se 1	plăcut 1
este 2	întâmplă 1	şi 1
mai 1	bineînţeles 1	vara 1
frig 1	pentru 1	

162. Alfabetul Morse codifică fiecare literă a alfabetului englez printr-un şir de puncte şi linii, astfel:

A .-	N -.
B -...	O ---
C -.-.	P .--.
D -.	Q --.-
E	R .-.
F ..-.	S
G --.	T -
H	U ..-
I	V .-
J .---	W .—
K -.-	X -..-
L .-..	Y -.—
M -	Z --..

Mesajul codificat Morse este reprezentat printr-un şir de biţi, după regulile:

a) este codificat prin `1`

- este codificat prin `111`

Oricare două coduri consecutive sunt separate printr-un `0`.

Exemplu. `M` este reprezentat prin `1110111`, iar `B` prin `111010101`

b) Literele interioare aparţinând aceluiaşi cuvânt sunt separate prin `000`.

c) Cuvintele sunt separate prin `00000`.

Exemplu. **ALB ROSU** se codifică prin:

```
1011100010111010100011101010100000101110100011101110111000101010001010111
```

Fişierul text '**MORSE.IN**' conţine una sau mai multe linii. Fiecare linie conţine o succesiune de **0** şi **1**. Primul **1** de pe linie poate fi precedat de o serie de **0** nesemnificativi. Fiecare linie se termină cu **7** de **0**. Dacă un caracter de pe o linie nu respectă una din regulile anterioare, linia se decodifică până la apariţia primei erori, după care se scrie '**?**'

Fiind dat un text în cod Morse în fişierul text '**MORSE.IN**', să se scrie un program care scrie textul decodificat ***cu majuscule*** în fişierul text de ieşire '**MORSE.OUT**'. Cuvintele vor fi separate printr-un singur spaţiu, fără semne de punctuaţie.

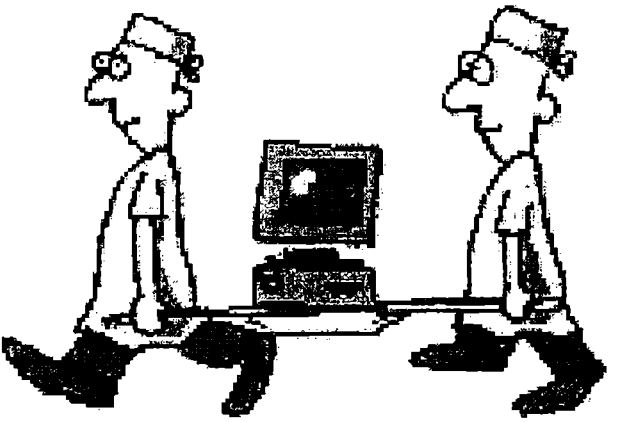
Lungimea maximă a unei linii din fişierul de intrare este **240**.

Exemplu. Dacă fişierul '**MORSE.IN**' este:

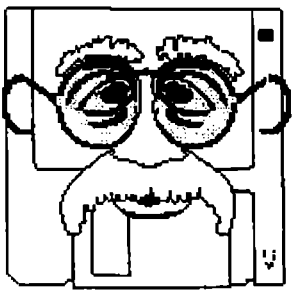
```
00000000101.1000111010100000011_0101110100000000
00110000000
000101110000000
0000000010111000111010100000111010111010000000
1110101000100010101110100010100000000
```

Atunci fişierul de ieşire '**MORSE.OUT**' va fi:

```
AD
?
A
AD C
DEFI
```



5. Subprograme



1. Să se scrie un program care ordonează crescător elementele de pe diagonala unei matrice `A` cu `n` linii şi `n` coloane prin interschimbarea liniilor şi coloanelor matricei.

   Exemplu. Matricea $\begin{pmatrix} 4 & 5 & 2 \\ 1 & 9 & 7 \\ 8 & 6 & 3 \end{pmatrix}$ devine $\begin{pmatrix} 3 & 8 & 6 \\ 2 & 4 & 5 \\ 7 & 1 & 9 \end{pmatrix}$.

2. Să se tipărească toate numerele prime din intervalul `[a,b]`.

   Exemplu. Pentru `a=15` şi `b=30` se vor afişa valorile `17,19,23,29`.

3. Să se tipărească numerele din intervalul `[a,b]` ce se divid cu suma cifrelor lor.

   Exemplu. Pentru `a=20`, `b=50` se afişează valorile `20,21,24,27,30,36, 40,42,45,48,50`.

4. Dintre primele `n` numere naturale, să se afişeze acelea cu proprietatea că suma cifrelor lor este impară. Se va folosi o funcţie care returnează suma cifrelor unui număr natural dat ca parametru.

   Exemplu. Dacă `n=30` se obţin numerele `1,3,5,7,9,10,12,14,16,18, 21,23,25,27,29`.

5. Scrieţi o funcţie care verifică dacă un număr este palindrom sau nu. Funcţia va avea:

   a) un parametru de tip integer;

   b) un parametru de tip string.

   Exemplu. `12321` este palindrom.

6. Să se scrie o funcţie care determină a `m`-a cifră a numărului `n`. Numerotarea se face de la stânga începând cu `0`.

   Exemplu. Dacă `n=31724` şi `m=3` se afişează cifra 7. 2

7. Să se scrie un program care modifică un vector astfel încât el să conţină doar valori distincte. Se va folosi un subprogram care verifică dacă o valoare există deja în vector.

   Exemplu. `x=(1,5,2,1,7,9,5,4,1)` => `x=(1,5,2,7,9,4)`.

8. Se dă matricea `A` cu `m` linii şi `n` coloane. Să se permute circular spre dreapta cu `k` poziţii toate liniile matricei. Se va folosi un subprogram care permută circular spre dreapta cu o poziţie o linie a matricei.

Exemplu. Dacă matricea este $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix}$ şi `k=2` rezultă $\begin{pmatrix} 3 & 4 & 1 & 2 \\ 7 & 8 & 5 & 6 \end{pmatrix}$.

9. Se citeşte de la tastatură o matrice `A` cu `m` linii şi `n` coloane şi elemente numere întregi. Să se determine linia (liniile) din matrice care conţine cele mai multe elemente nenule. Se va folosi o funcţie care returnează numărul elementelor nenule de pe o linie a cărui indice `i` se transmite ca parametru.

Exemplu. Pentru matricea $\begin{pmatrix} 1 & 0 & 2 & 0 \\ 5 & 0 & 4 & 9 \\ 7 & 0 & 0 & 1 \end{pmatrix}$ liniile cerute sunt `1` şi `3`.

10. Se dă un vector `x` cu `n` componente numere întregi. Să se calculeze:

    a) suma componentelor din vectorul `x`;

    b) numărul componentelor impare din `x`;

    c) elementul maxim din vector.

    Exemplu. Pentru `x=(2,9,15,6,3,12,66,8,5)` => a) `126`; b) `4`; c) `66`.

11. Elementele unei matrice `A` cu `m` linii şi `2` coloane sunt coordonatele a `m` puncte în plan $P_1, P_2, \ldots, P_m$, $m \geq 3$. Să se afişeze numerele de ordine ale celor trei linii care determină triunghiul de arie maximă, precum şi valoarea acestei arii. Se va folosi o funcţie care returnează aria unui triunghi de laturi date.

12. Se dă un vector cu numere întregi. Să se afişeze toate perechile de numere prime între ele. Se va folosi o funcţie care determină cel mai mare divizor comun a două numere.

13. Se dau $m, n \in N$ şi mulţimile $A = \{a_1, a_2, \ldots, a_m\}$ şi $B = \{b_1, b_2, \ldots, b_n\}$ memorate ca vectori. Să se scrie câte un program care calculează:

    a) $C = A \cap B$

    b) $C = A \cup B$

    c) $C = A - B$

    Vectorii `A` şi `B` se consideră nesortaţi. Se va folosi o funcţie care verifică dacă o anumită valoare se găseşte într-un vector sau nu.

    Exemplu. Dacă `A=(1,5,2,9,12)` şi `B=(5,3,4,6,33,99,12,64)` atunci a) `C=(5,12)`; b) `C=(1,5,2,9,12,3,4,6,33,99,64)`; c) `C=(1,2,9)`.

14. Se consideră doi vectori `A` cu `m` elemente și `B` cu `n` elemente. Se cere să se determine reuniunea celor doi vectori folosind:

    un subprogram pentru ordonarea unui vector;

    un subprogram pentru a insera o valoare într-un vector ordonat pentru a rămâne ordonat (se va folosi căutarea binară a poziției de inserare).

    Exemplu. Dacă `A=(1,5,2,9,12)` și `B=(5,3,4,6,33,99,12,64)` atunci `A` ∪ `B=(1,5,2,9,12,3,4,6,33,99,64)`.

15. Se citește de la tastatură un șir de caractere. Să se verifice dacă acesta conține doar caractere mici sau mari ale alfabetului englez și în caz afirmativ afișează mulțimea literelor distincte din șirul dat.

    Exemplu. Pentru șirul `'acesta este un sir'` răspunsul este afirmativ iar mulțimea solicitată este `'a','c','e','i','n','r','s','t','u'`

16. Numim linii gemene într-o matrice, două linii care conțin exact aceleași elemente, eventual într-o altă dispunere pe coloane. Scrieți un program care determină toate perechile de linii gemene într-o matrice `A` dată, cu `m` linii și `n` coloane, și elemente numere întregi. Pentru fiecare astfel de pereche, se vor afișa pe un rând indicii liniilor componente.

    Exemplu. $A = \begin{pmatrix} 1 & 5 & 2 & 4 \\ 3 & 9 & 7 & 2 \\ 2 & 5 & 4 & 1 \end{pmatrix}$ singura pereche este `(1,3)`

17. Se dă o matrice `A` cu `m` linii și `n` coloane. Să se elimine din `A` liniile și coloanele la intersecția cărora se află un număr natural al cărui scriere în baza `2` are cel puțin două cifre egale cu `1`. Se va folosi o funcție care returnează numărul de cifre egale cu `1` dintr-un număr, o procedură de eliminare a unei linii dintr-o matrice și o procedură care elimină o coloană dintr-o matrice.

18. Se introduce de la tastatură o cifră `k` diferită de `0` și `1` și un text `x` cu cel mult `255` caractere, format din litere mici ale alfabetului englez. Textul `x` se codifică literă cu literă, fiecare literă fiind înlocuită cu un număr natural egal cu k^{poz}, unde `poz` este poziția literei în alfabet. Astfel `'a'` se codifică prin k^1, `'b'` se codifică prin k^2, ș.a.m.d. Între numerele codificării se lasă un spațiu liber. Se cere să se afișeze codificarea textului. Se va folosi o funcție care returnează puterea a^b, și o funcție care returnează poziția unei litere în cadrul alfabetului.

19. Se consideră funcția `f` definită prin:

$$f(x,y) = \begin{cases} x^2 + y^2 & \text{daca } \max(|x|,|y|) < 1 \\ \frac{1}{x} + \frac{1}{y} & \text{daca } \min(|x|,|y|) > 1. \\ x + y & \text{altfel} \end{cases}$$

    Se citesc `n` perechi `(x,y)`. Să se afișeze cea mai mare respectiv cea mai mică valoare a funcției `f`.

20. Scrieţi un program care, pentru doi întregi `x` şi `n`, afişează valoarea expresiei

$$S = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \quad + \frac{x^{2n-1}}{(2n-1)!}.$$

Se vor folosi trei subprograme: o funcţie pentru ridicarea la putere, o funcţie de calcul al factorialului şi o funcţie care primind ca parametrii valorile `x` şi `n`, returnează valoarea expresiei `S`.

21. Să se scrie un program care determină suma componentelor prime dintr-un vector de numere întregi. Se va folosi o funcţie care verifică dacă un număr este prim sau nu.

Exemplu. Pentru vectorul `x=(5,9,2,8,18,3)` suma cerută este `s=10`.

22. Se dă un vector de `n` numere întregi. Să se afişeze numerele din vector care au cele mai multe cifre egale cu `c` (citit de la tastatură) în scrierea lor în baza `b` (`b<=9`). Se va folosi o funcţie care returnează numărul de cifre `c` din scrierea în baza `b` a unui număr.

23. Se citeşte de la tastatură un text terminat format din litere mici ale alfabetului englez precum şi separatorii punct (.), virgulă (,), punct şi virgulă (;), spaţiu şi două puncte (:). Se cere să se scrie un program care formează toate grupurile de cuvinte în care fiecare cuvânt din grup reprezintă o anagramă a celorlalte cuvinte din grup. Nu se vor afişa grupurile dintr-un singur cuvânt. Se va folosi o procedură care returnează primul cuvânt dintr-un text şi o funcţie care verifică dacă un cuvânt este anagrama altui cuvânt. (se pot folosi şi alte funcţii sau proceduri!)

24. Se citesc `n` numere naturale de la tastatură. Să se afişeze numerele din vector pe grupe, o grupă având numere cu aceeaşi sumă de control. Grupele se vor afişa în ordinea descrescătoare a numărului de elemente din grupă. Dacă două grupe au acelaşi număr de componente ele se vor afişa în ordinea crescătoare a valorii cifrei de control corespunzătoare grupei respective. Suma de control a unui număr se obţine calculând suma cifrelor numărului, apoi suma cifrelor acestei cifre ş.a.m.d. până se obţine o sumă formată dintr-o singură cifră care va reprezenta cifra de control.

Exemplu. Pentru `x=(156,91,12,10081,822,75)` se va afişa:

Grupa 1: `156, 12, 822, 75`

Grupa 2: `10081`

Grupa 3: `91`.

25. Să se scrie un program care tipăreşte un număr precizat de termeni din şirul `1, 2, 1, 3, 2, 1, 4, 2, 2, 5, 4, 3, 2, 1, 6, 2, 2, 3, 3, 3...` obţinut din şirul numerelor naturale prin înlocuirea fiecărui număr natural printr-un grup de numere. Numărul prim `p` este înlocuit prin numerele `p, p-1, ... , 3, 2, 1` iar numărul compus `n` este înlocuit prin `n` urmat de toţi divizorii săi proprii, un divizor `d` repetându-se de `d` ori.

26. Realizați, folosind tablouri, o aplicație practică referitoare la un concurs de admitere într-un liceu cu un singur profil. Există n elevi candidați, iar numărul de locuri este m. Fiecare elev ia două note. Primii m elevi sunt considerați admiși dacă fiecare notă a lor este mai mare sau cel puțin egală cu 5. Aplicația va permite, așadar, ordonarea elevilor după media obținută, dar se cer și o ordonare alfabetică, precum și implementarea căutării unui elev după nume. Pentru fiecare din aceste operații, precum și pentru citirea datelor și afișarea diferitelor situații se vor implementa câte o funcție sau procedură. Alegerea opțiunilor se va face de către utilizator prin intermediul unui mic "meniu".

27. Un număr natural este perfect dacă este egal cu suma divizorilor săi diferiți de el însuși. Să se determine toate numerele perfecte mai mici decât n, citit de la tastatură.

    Se vor folosi următoarele proceduri și funcții:

    procedură de determinare a divizorilor unui număr întreg

    funcție booleană care verifică dacă un număr este perfect sau nu.

    Exemplu. Dacă n=500 se afișează valorile 6,28,496.

28. Dându-se un vector neordonat A cu n elemente, să se găsească cel de-al k-lea element din tabloul ordonat, fără a-l ordona. Se va folosi o funcție care determină poziția minimului elementelor unui șir.

    Indicație. Determinarea minimului șirului dat realizează de fapt găsirea primului termen al șirului ordonat fără a-l ordona. Dacă pe poziția acestui minim punem o valoare mai mare decât a tuturor termenilor, la următorul pas determinarea minimului va realiza stabilirea ce lui de-al doilea termen al șirului ordonat ș.a.m.d.

29. Printre numerele mai mici sau egale cu un număr n dat, să se găsească cel care are mai mulți divizori. Se va folosi o funcție ce calculează numărul de divizori ai unui număr.

    Exemplu. Pentru n=100, numărul cerut este 60 care are 12 divizori.

30. Să se calculeze cel mai mare divizor comun a două numere date m și n astfel:

    pentru fiecare număr se creează un tablou cu toți divizorii acestuia inclusiv divizorii banali (1 si numărul însuși)

    se selectează apoi într-un tablou toți divizorii comuni și se ia cel mai mare dintre aceștia

    Se va folosi programarea modulară.

31. Numerele puternice sunt acele numere naturale care au numărul divizorilor mai mare decât numărul divizorilor oricărui număr natural mai mic decât numărul respectiv. Programul cere să se afișeze al n-lea număr puternic (n<70).

    Exemplu. Cel mai mic număr puternic este 1, urmează 2, 4 și 6.

32. Să se scrie o procedură de tip funcție **max2** care determină maximul dintre valorile a două variabile de tip întreg, apoi să se scrie o procedura de tip funcție **max3** care determină maximul dintre valorile a trei variabile de tip întreg astfel încât să utilizeze funcția **max2**.

33. Să se scrie o procedură **perm2** care realizează interschimbarea valorilor a două variabile și să se scrie apoi o procedură **ord** care ordonează crescător un șir de numere întregi astfel încât să apeleze **perm2**.

34. Scrieți un program care să afișeze calendarul pe un an, dându-se anul și ziua din săptămână în care cade `1` Ianuarie. Veți folosi o procedură care să afișeze calendarul pentru o lună, dându-se ziua din săptămână cu care o lună începe și numărul de zile din lună. Fiecare lună succesivă începe cu ziua din luna ce urmează după ultima zi a lunii precedente. Zilele săptămânii vor fi numerotate de la `0` la `6` pentru Duminică până la Sâmbătă. Nu uita de anii bisecți.

35. Se dă un tablou bidimensional cu `m` linii și `n` coloane având componente numere naturale. Să se determine cel mai mare divizor comun și cel mai mic multiplu comun al componentelor tabloului.

36. Pentru un număr natural `n` dat, să se stabilească raportul dintre numărul divizorilor pari și numărul divizorilor impari ai numărului `n`.

    Exemplu. Pentru `n=246` raportul este `1` (`4` divizori pari și `4` divizori impari).

37. Se citesc de la tastatură `k` matrice cu `m` linii și `n` coloane. Să se determine câte din aceste matrice verifică cel puțin una din următoarele proprietăți:

    a) există cel puțin o linie cu toate elementele pozitive;

    b) există cel puțin o linie cu toate elementele în progresie geometrică;

    c) există cel puțin o linie cu toate elementele ordonate crescător.

38. Să se rezolve în mulțimea numerelor reale ecuația $ax^4+bx^2+c=0$, folosind un subprogram pentru rezolvarea ecuației de gradul II.

39. Să se determine punctele șa dintr-o matrice `A` cu `m` linii și `n` coloane folosind un subprogram care determină maximul dintr-un tablou liniar și un subprogram care determină minimul dintr-un tablou liniar. (*Observație.* elementul a_{ij} se numește element șa a matricei `A` dacă el este minim pe linia `i` și maxim pe coloana `j` sau invers)

40. Să se scrie subprograme pentru lucrul cu șiruri de caractere, presupunând că nu există funcțiile sistem **length**, **pos**, **copy**, **delete**, **insert**, **str**, **val**.

41. Există numere care au proprietatea că se scriu, în două baze diferite, prin trei cifre identice. De exemplu, numărul 273_{10} în baza `9` se scrie `333`, și în baza `16` se scrie `111`. Să se scrie un program care să determine toate numerele mai mici ca `N` care au această proprietate.

42. Se dau n mulțimi cu componente numere întregi, memorate într-o matrice cu 2*n linii astfel: fiecare mulțime este dată prin numărul de elemente pe o linie a matricii iar elementele mulțimii pe linia următoare a matricei. Se cere să se afle elementele intersecției acestor mulțimi. Se va folosi o funcție care caută un număr pe o linie a matricii, și o procedură care determină intersecția a două linii a matricei.

Exemplu. Pentru n=3 și matricea

4	0	0	0	0
1	2	3	4	0
3	0	0	0	0
8	1	4	0	0
5	0	0	0	0
7	8	1	3	4

se va afișa: 1, 4.

43. Scrieți un program care verifică relația

$$cmmdc(f_n, f_m) = f_{cmmdc(n,m)}$$

pentru două numere întregi m și n citite de la tastatură. Prin f_k s-a notat al k-lea element al șirului lui Fibonacci definit astfel:

$$f_0 = 1; \qquad f_1=1; \qquad f_n = f_{n-1} + f_{n-2}, \quad n>=2.$$

44. Se dă un polinom P de grad n și m valori reale $a_1, a_2, \ldots, a_m$. Să se calculeze valoarea polinomului pentru cele m valori citite precum și maximul celor m valori $P(a_i)$.

45. Un centru meteo are n angajați. Fiecare angajat petrece la serviciu o anumită perioadă din zi, delimitată de momentele sosirii și plecării. În orice moment al zilei este prezent cel puțin un angajat. Să se determine perioada în care la centrul meteo se află un număr maxim de angajați.

46. Spunem despre un număr că are proprietatea sufixului, dacă el apare ca sufix al pătratului său, adică cifrele sale apar la sfârșitul șirului de cifre ce reprezintă acel număr ridicat la pătrat. De exemplu 9376 are proprietatea sufixului în baza 10 ($9376^2=87909376$). Este clar că baza în care lucrăm este importantă, un număr poate avea proprietatea sufixului într-o bază și poate să nu o aibă în altă bază. Scrieți un program care citește de la tastatură un număr natural n (scris în baza 10) și verifică dacă numărul are proprietatea sufixului în vreo bază b<=10. Se va folosi o funcție/procedură de transformare a unui număr din baza 10 în baza b, o funcție/procedură care verifică dacă un număr are proprietatea sufixului într-o bază b.

47. Să se calculeze suma $2^{x[1]}+2^{x[2]}+\ldots+2^{x[n]}$, pentru n număr natural și x=(x[1],x[2],...,x[n]) vector de numere întregi citite de la tastatură.

Observație. Valorile obținute pot fi foarte mari. De exemplu 2^{100} conține 31 de cifre.

Exemplu. Pentru n=5 și x=(5,10,35,12,22) se obține suma 34363937824.

48. Se dau două numere `a` şi `b` memorate ca şiruri de caractere. Să se calculeze şi afişeze suma şi produsul celor două numere.

49. Se citesc de la tastatură `n` numere naturale cu cel mult `255` cifre. Să se afişeze numărul care are suma cifrelor cea mai mare. Se va folosi o funcţie care are ca parametru un astfel de număr (memorat ca şir de caractere) şi returnează suma cifrelor acestui parametru.

50. Să se scrie câte un subprogram pentru a calcula:

    a) ultima cifră a numărului 2^x, pentru `x` număr natural;

    b) ultima cifră a numărului a^b, cu `a` şi `b` cifre ale sistemului zecimal;

    c) ultima cifră a sumei $2^a+3^a+...+9^a$, cu `a` număr natural;

    d) ultima cifră a numărului a^b, pentru `a` şi `b` numere naturale mai mici sau egale cu `32000`, citite de la tastatură;

    e) ultima cifră a sumei $1^n+2^n+...+n^n$, pentru `n` număr întreg citit de la tastatură.

    Exemplu. a) `x=15 => 8`; b) `a=3, b=9 => 3`; c) `a=11 => 4`;

    d) `a=123, b=547 => 7`; e) `n=11 => 6`.

51. Se dă o matrice pătratică `A`, de dimensiune `n`. Să se scrie un program care afişează matricele: A, A^2, A^3, ..., A^n.

52. Pentru codificarea unui text format din litere mici se folosesc două tablouri bidimensionale `5×5`, generate prin program.

a	b	c	d	e
f	g	h	i	j
k	l	m	n	p
o	r	s	t	u
v	w	x	y	z

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Codificarea se realizează caracter după caracter astfel:

caracterul din linia `i` şi coloana `j` se codifică prin elementul corespunzător din tabloul de numere întregi;

după efectuarea acestei codificări, linia `i` din tabloul de numere se permută circular spre dreapta cu `i` poziţii, apoi coloana `j` din acelaşi tabel se permută circular în sus cu `j` poziţii;

Să se scrie un program care, pentru orice text dat conţinând doar caractere mici ale alfabetului englez, să afişeze succesiunea de numere obţinută prin codificare. Textul poate să conţină şi spaţii care se vor ignora. Între două coduri succesive se va lăsa exact un spaţiu.

Exemplu. Codificarea şirului `'exemplu'` este `5 23 4 2 12 11 20`.

53. Simulaţi jocul de `Bingo`. Jucătorul introduce o variantă jucată, iar calculatorul generează aleator varianta câştigătoare şi afişează rezultatul (linie, bingo, nimic). Varianta jucată şi cea câştigătoare se vor afişa pe ecran colorate, punându-se în evidenţă cu culori intermitente linia sau cartonul câştigător, dacă varianta jucată este câştigătoare.

54. Se consideră o matrice rară (adică cu foarte multe elemente nule) `A(n,n)` cu `p` elemente nenule. Matricea este memorată economic sub forma a doi vectori `V(i)`, `T(i)`, $1 \le i \le p$, care reţin valorile elementelor nenule ale matricei, respectiv poziţia acestor elemente, liniarizate pe linii.

Să se scrie un program care realizează următoarele operaţii, fiecare fiind implementată cu ajutorul unui subprogram separat:

a) citind vectorii `V` şi `T` reface matricea `A`;

Exemplu. Pentru `n=5`, `p=8`, `V=(2,7,8,4,6,1,3,5)`, `T=(1,5,9,11,14,18,20,22)` se va afişa matricea $A = \begin{pmatrix} 2 & 0 & 0 & 0 & 7 \\ 0 & 0 & 0 & 8 & 0 \\ 4 & 0 & 0 & 6 & 0 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 5 & 0 & 0 & 0 \end{pmatrix}$.

b) determină vectorii `V` şi `T` corespunzători matricei transpuse a matricei `A`, fără a reconstitui matricea;

Exemplu. Transpusa matricei anterioare este dată prin `n=5`, `V=(2,4,5,1,8,6,7,3)` şi `T=(1,3,10,14,17,18,21,24)`.

c) considerând matricele `A(n,n)` şi `B(n,n)` cu `p` respectiv `q` elemente nenule, calculează suma celor două matrice folosind memorarea economică;

Exemplu. Pentru `n=5` şi matricele date prin `V1=(2,7,8,4,6,1,3,5)`, `T1=(1,5,9,11,14,18,20,22)` şi respectiv `V2=(1,5,1,3,4,4,2,9)`, `T2=(1,2,8,10,12,16,18,24)` se obţine `V=(3,5,7,1,8,3,4,4,6,4,3,3,5,9)` şi `T=(1,2,5,8,9,10,11,12,14,16,18,20,22,24)`.

d) acelaşi enunţ ca la punctul anterior pentru produsul celor două matrice.

Exemplu. Pentru `n=5` şi matricele date prin `V1=(2,7,8,4,6,1,3,5)`, `T1=(1,5,9,11,14,18,20,22)` şi respectiv `V2=(1,5,1,3,4,4,2,9)`, `T2=(1,2,8,10,12,16,18,24)` se obţine `V=(2,10,63,32,16,28,20,12,4,27,5,15)` şi `T=(1,2,4,6,8,11,12,13,17,19,23,25)`.

Observaţie. La toate punctele se consideră că elementele nenule se vor citi/afişa în ordinea liniarizării pe linii a matricei (adică în ordinea $a_{11}, a_{12}, \ldots, a_{1n}, a_{21}, \ldots, a_{2n}, \ldots, a_{nn}$, deci vectorul `T` este sortat crescător).

55. Se consideră o matrice `A(n,n)` cu `p` elemente nenule. Matricea este memorată economic sub forma a trei vectori `V(i)`, `L(i)` şi `C(i)` care reţin valorile, liniile şi coloanele elementelor nenule.

Să se scrie un program care realizează următoarele operaţii fiecare printr-un subprogram:

a) citind vectorii `V`, `L`, `C` reface matricea `A`;

Exemplu. Pentru `n=5`, `V=(1,5,1,3,4,4,2,9)`, `L=(1,1,2,2,3,4,4,5)`, `C=(1,2,3,5,2,1,3,4)` se obţine matricea $\begin{pmatrix} 1 & 5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 4 & 0 & 0 & 0 \\ 4 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 \end{pmatrix}$.

b) considerând matricele `A(n,n)` şi `B(n,n)` cu `p` şi respectiv `q` elemente nenule, calculează suma celor două matrice folosind memorarea economică.

Exemplu. Pentru `n=5` şi `V1=(1,5,1,3,4,4,2,8)`, `L1=(1,1,2,2,3,4,4,5)`, `C1=(1,2,3,5,2,1,3,4)`; respectiv `V2=(2,7,8,4,6,1,3,5)`, `L2=(1,1,2,3,3,4,4,5)`, `C2=(1,5,4,1,4,3,5,2)` se obţin vectorii `V=(3,5,7,1,8,3,4,4,6,4,3,3,5,8)`, `L=(1,1,1,2,2,2,3,3,3,4,4,4,5,5)` şi `C=(1,2,5,3,4,5,1,2,4,1,3,5,2,3)` corespunzători matricei sumă.

c) acelaşi enunţ pentru produsul celor două matrice.

Exemplu. Pentru `n=5` şi `V1=(1,5,1,3,4,4,2,8)`, `L1=(1,1,2,2,3,4,4,5)`, `C1=(1,2,3,5,2,1,3,4)`; respectiv `V2=(2,7,8,4,6,1,3,5)`, `L2=(1,1,2,3,3,4,4,5)`, `C2=(1,5,4,1,4,3,5,2)` se obţin vectorii `V=(2,40,7,4,15,6,32,16,12,28,8,24)`, `L=(1,1,1,2,2,2,3,4,4,4,5,5)` şi `C=(1,4,5,1,2,4,4,1,4,5,3,5)` corespunzători matricei produs.

Observaţie. La toate punctele se consideră că elementele nenule se vor citi/afişa în ordinea liniarizării pe linii a matricei (adică în ordinea a_{11}, a_{12}, ... a_{1n}, a_{21}, ... a_{2n}, ..., a_{nn}, deci vectorul `T` este sortat crescător).

56. Un panou de tir distractiv este format din `n×n` casete ce ascund beculeţe legate în serie pe trasee care descriu diverse figuri. O casetă poate să conţină mai multe beculeţe, fiecare beculeţ aparţinând unei figuri distincte. Se consideră că ţintirea unei casete acţionează asupra întrerupătoarelor circuitelor electrice corespunzătoare traseelor pe care sunt înserate beculeţele din casetă.

Cunoscând traseele care compun fiecare figură, să se determine numărul minim de casete, şi poziţiile acestora, astfel încât să se obţină luminarea tuturor figurilor.

57. Se dă fişierul text `DOC.TXT`, care conţine o documentaţie. Se cere crearea unui fişier care să conţină o tabelă de indecşi ordonată crescător, formată din cuvintele folosite în documentaţie. O linie de index va avea următoarea structură:

```
cuvânt      NL1, [NA1]
            NL2, [NA2],

            NLn, [NAn]
```

unde `Nli` reprezintă numărul liniei în care apare cuvântul, `Nai` reprezintă numărul de apariţii ale cuvântului pe linia `Nli`.

Observaţie. Se ignoră diferenţele dintre literele mari şi mici, iar cuvintele sunt separate prin unul sau mai multe caractere , . spaţiu.

58. Să se scrie un program care citeşte de la tastatură două numere întregi `x` şi `y` şi afişează rezultatele următoarelor operaţii:

şi pe biţi

sau pe biţi

sau exclusiv pe biţi

negare pe biţi

Pentru aceasta se va scrie câte o funcţie pentru fiecare operaţie şi în plus o procedură de reprezentare în baza 2 a unui număr întreg. Această reprezentare se va memora într-un vector de tip `array [1..16] of 0..1`.

Exemplu. Dacă `x=736 y=297`

$\Rightarrow$ $x=1011100000_2$ $y=100101011_2$

1011100000

0100101011

`x si pe biţi y` $0000100000_2 = 32$

`x sau pe biţi y` $1111101011_2 = 1003$

`x sau ex. pe biţi y` $1111001011_2 = 971$

`not x` $0100011111_2 = 287$

59. Se consideră funcţia $f \quad N \rightarrow N$, $f(n) = n - c_{\pi(n)}$, unde c_k este al k-lea număr prim, iar $\pi(k)$ reprezintă numărul numerelor prime mai mici sau egale cu k (indicatorul lui Euler). Fiind dat un număr M, să se elaboreze un program care determină $\max_{2 \le i \le M}\{f(i)\}$ şi numărul i pentru care este atinsă această valoare.

60. Linia a i-a a unei matrice A cu m linii şi n coloane (m<=n) se notează cu $a_{i\cdot}$ şi se interpretează ca un vector cu n elemente. Să se construiască, prin intermediul unei matrice C de tip m×n, o matrice B de tip m×n ale cărei linii se obţin din A astfel:

$$b_{k\cdot} = \frac{1}{\|c_{k\cdot}\|} c_{k\cdot} \text{ unde}$$

$$c_{k\cdot} = a_{k\cdot} - \sum_{i=1}^{n-1} (a_{k\cdot}, b_{i\cdot}) b_{i\cdot}$$

$$(a_{k\cdot}, b_{i\cdot}) = \sum_{j=1}^{n} a_{kj} b_{ij}$$

$$\|c_{k\cdot}\| = \sqrt{(c_{k\cdot}, c_{k\cdot})}.$$

61. O colonie este formată din n celule aşezate pe un cerc; unele celule sunt vii şi altele moarte. Definim evoluţia cu o generaţie a coloniei ca fiind procesul prin care toate celulele îşi schimbă simultan starea conform următoarelor reguli:

    între două celule moarte nu poate exista o celulă vie;

    între două celule vii, orice celulă vie se sufocă şi moare;

    între o celulă moartă şi una vie, o celulă supravieţuieşte, sau dacă era moartă, una nouă se naşte în loc.

    Evident, starea unei celule din generaţia k depinde exclusiv de stările celulelor vecine ei din generaţia k-1.

    Spunem că o colonie se stabilizează dacă există o generaţie S>=0 începând de la care nici o celulă nu îşi schimbă starea, adică aspectul coloniei este acelaşi la generaţiile S, S+1, S+2,...

    Dându-se o colonie cu N celule, să se spună dacă ea se stabilizează şi dacă da, care este valoarea minimă pentru S (minimă în sensul că generaţia S-1 este diferită de generaţia S).

62. Pentru calculul duratei de viaţă a unui anumit tip de piesă se cunosc date observate la diferite momente de timp T, prezentate sub forma unui tabel ca mai jos:

T (ani)	0	1	2	3		Q-1	Q
N (T)	1000	1000	900	850	...	40	0

în care T indică momentele de timp, iar N(T) numărul de bucăţi rămase în funcţiune la momentul T>0 faţă de lotul iniţial de piese (în exemplul de mai sus de 1000). Să se scrie programul pentru calculul următorilor indicatori:

probabilitatea de avarie la diferite momente de timp T>0, după relaţia:

$$P(T) = \frac{N(T-1) - N(T)}{N(0)}$$

probabilitatea condiţionată de avarie, după relaţia:

$$PC(T) = 1 - \frac{N(T)}{N(T-1)}$$

durata medie de viaţă a unei piese, după relaţia:

$$DV = \sum_{T=1}^{Q} P(T) * T$$

Rezultatele se vor prezenta pe ecran sub forma unui tabel ca cel de mai jos:

TIMP SCURS (T)	RAMAS IN FUNCTIUNE N(T)	MORTALITATE N(T-1)-N(T)	PROBABIL. DE AVARIE P(T)	PROBABIL. CONDITIONATA PC(T)
0				
1				
2				
Q				

Durata medie de viata =

63. Scrieţi un program care joacă tic-tac-toe (X şi 0). Reprezentaţi tabla ca un tablou de caractere 3 × 3. Iniţializaţi tabloul cu spaţii şi cereţi fiecui jucător pe rând să introducă poziţia. Poziţia primului jucător va fi marcată pe tablă cu un '0' iar poziţia celui de al doilea jucător va fi marcată cu un 'X'. Continuaţi procesul până când un jucător câştigă sau se ajunge la remiză.

Poziţia fiecărui jucător va fi introdusă sub forma: număr linie, spaţiu, număr coloană. După fiecare joc afişaţi o diagramă cu tabla de joc arătând poziţia finală. Păstraţi un contor al numărului de jocuri câştigate de fiecare jucător şi al jocurilor remiză.

Înainte de a începe fiecare joc întrebaţi fiecare jucător dacă vrea să continue. Dacă unul, dintre ei nu mai doreşte, afişaţi statistica jocului şi terminaţi programul.

64. Valoarea factorialului creşte foarte repede. Pentru a reprezenta factorialele vom utiliza descompunerea în factori primi. Să se scrie un program care afişează pentru un n citit de la tastatură, descompunerea în factori primi a lui n!. Nu se va efectua calculul efectiv al factorialului!

65. Seiful unei bănci are ca cifru un număr natural N. Cheia electronică folosită pentru deschiderea acestuia este un număr natural mai mic decât N. Directorul băncii scrie pe o foaie toate numerele pe care le foloseşte, dar uită care dintre ele este cheia seifului. Cheia care deschide seiful are o reprezentare binară care, prin suprapunere peste o secvenţă din descompunerea binară a cifrului lacătului, modifică cifrul după următoarele reguli:

0 peste 1 nu modifică cifra;

1 peste 0 transformă cifra în 1;

1 peste 1 transformă cifra în 0;

0 peste 0 nu modifică cifra.

Lacătul se deschide dacă reprezentarea binară a cifrului lacătului, după suprapunere, conţine doar cifre binare cu valoarea 1.

Ajutaţi-l pe director să caute printre numerele date cheile corecte.

Fişierul de intrare SEIF.IN conţine pe prima linie cifrul seifului. Pe a doua linie se află numărul numerelor de pe foaie. Pe a treia linie se găsesc numerele de pe foaie, separate printr-un singur spaţiu.

Fişierul de ieşire CUVINTE.OUT va conţine o singură linie pe care se vor afla cheile corecte, separate prin spaţii.

66. O firmă de hardware îşi codifică produsele astfel: un cod este format dintr-o secvenţă de cifre urmată de o secvenţă caractere (litere mici ale alfabetului englez). În momentul apariţiei unui nou produs, firma este interesată de codul următorului produs şi de codul produsului anterior. Codurile produselor sunt generate astfel: se generează secvenţa următoare în ordine lexicografică formată din aceleaşi caractere ale codului produsului anterior (fără a modifica secvenţa numerică a codului anterior), iar dacă nu există o secvenţă următoare se incrementează cu 1 secvenţa numerică şi se generează primul cod în ordine lexicografică, format din aceleaşi caractere.

Se cere ca pentru un anumit cod dat, să se genereze codul anterior şi codul următor al unui produs. Dacă unul din aceste coduri nu există se va afişa * în locul său.

Exemplu. Pentru codul '000abc' se va afişa '*' şi '000acb', pentru '2457z' se va afişa '2456z' şi '2458z', pentru '153xabc' se va afişa '153cxba' şi '153xacb'

67. Reprezentarea Booth a permis accelerarea spectaculoasă a efectuării operaţiilor aritmetice în memoria internă. Codul Booth este o reprezentare a numărului în care baza este 3, dar cifrele sunt 0, 1, -1 (în loc de 2); vom reprezenta -1 prin !

Exemplu: 9 se reprezintă în cod Booth ca 100= $1*3^2+0*3^1+0*3^0$

14 ca 1!!=$1*3^3+(-1)*3^2+(-1)*3^1+(-1)*3^0$

Se cere:

a) Să se scrie o procedură care primind ca intrare reprezentarea Booth a unui număr natural oferă la ieşire reprezentarea acestuia în baza 10.

b) Să se scrie o procedură care primind ca intrare un număr natural reprezentat în baza 10, oferă la ieşire reprezentarea Booth a acestui număr.

c) Să se scrie o procedură care având la intrare reprezentarea Booth a două numere produce ca rezultate reprezentările sumei şi produsului celor 2 numere fără a efectua trecerea prin altă bază.

Indicaţii. Se ţine cont că

$$2 * 3^m = (3 - 1) * 3^m = 3^m + 1 + (-1) * 3^m$$

deci $2_{(3)} = 1(-1)_3 = 1!$

Exemplu.

$12_3 = 2 * (-1)_3 = 1(-1)(-1)_3 = 1!!$

$22_3 = 3(-1)_3 = 10(-1)_3 = 10!$

$1910 = 2121202_3 = 212121!_3 = 2122!1!_3 =$

$= 213!!1!_3 = 220!!1!_3 = 3!0!!1!_3 = 10!0!!1!_3$

Operaţiile de adunare şi înmulţire au următoarele reguli:

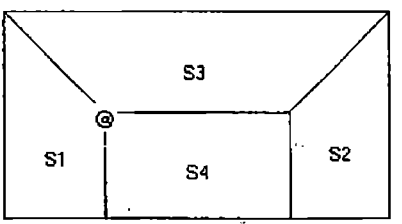
+	0	1	!
0	0	1	!
1	1	1!	0
!	!	0	!1

*	0	1	!
0	0	0	0
1	0	1	!
!	0	!	1

Exemplu.

```
  1!10!!110!+
  1!10!!1111
-------------
1!!!!1!00!0
```

68. Se consideră un tablou de numere întregi cu n linii şi n coloane. Dându-se poziţia (i,i), se împarte matricea în patru zone astfel:



Unde @ indică poziţia (i,i)

Să se determine: X – maximul dintre elementele pare interioare zonei S1; Y – minimul dintre elementele impare interioare zonei S2; Z – suma elementelor interioare zonei S3; P – produsul elementelor interioare zonei S4; R = Z/P.

69. Se ştie că în memoria calculatorului nu se pot memora întregi mai mari de o anumită valoare. Pentru a depăşi această limită se pot memora cifrele unui număr întreg într-un tablou de următoarea formă:

```
type intregExtins = array[1..p-1] of 0..9;
```

unde un întreg `x: intregExtins` va avea valoarea:

$$\text{val}(x) = \sum_{i=0}^{p-1} x[i] \cdot 10^{p-i-1}$$

Exemplu. `p=5`

$$\text{val}(x) = x[0] \cdot 10^4 + x[1] \cdot 10^3 + x[2] \cdot 10^2 + x[3] \cdot 10^1 + x[4] \cdot 10^0$$

`p` este dimensiunea maximală a reprezentării zecimale a unui număr întreg extins (`p` este declarat ca o constantă). Să se scrie câte o procedură pentru fiecare din operaţiile următoare:

a) adunarea a două numere întregi extinse;
b) determinarea succesorului unui număr întreg extins;
c) scăderea a două numere întregi extinse.

Indicaţie. $10^p = (0,0,\ldots,0)$ deci are aceeaşi reprezentare ca şi `0`.

$$\begin{aligned} a - b &= a - b + 0 = a - b + 10^p = \\ &= a + (10^p - b) = \\ &= a + succ(9 - b_0, 9 - b_1, \ldots, 9 - b_{p-1}) \end{aligned}$$

d) Împărţirea cu rest a două numere întregi extinse (Atenţie la împărţirea cu `0`);
e) Înmulţirea unui număr întreg extins cu o putere a lui `10`;
f) Înmulţirea unui număr întreg extins cu un număr între `0` şi `9`;
g) Înmulţirea a două numere întregi extinse.

70. Se consideră un pătrat de latură `1` cu colţurile în `(0,0)`, `(1,0)`, `(1,1)`, `(0,1)`. Împărţim acest pătrat în patru părţi egale şi numerotăm sferturile în ordinea următoare:

1	2
3	4

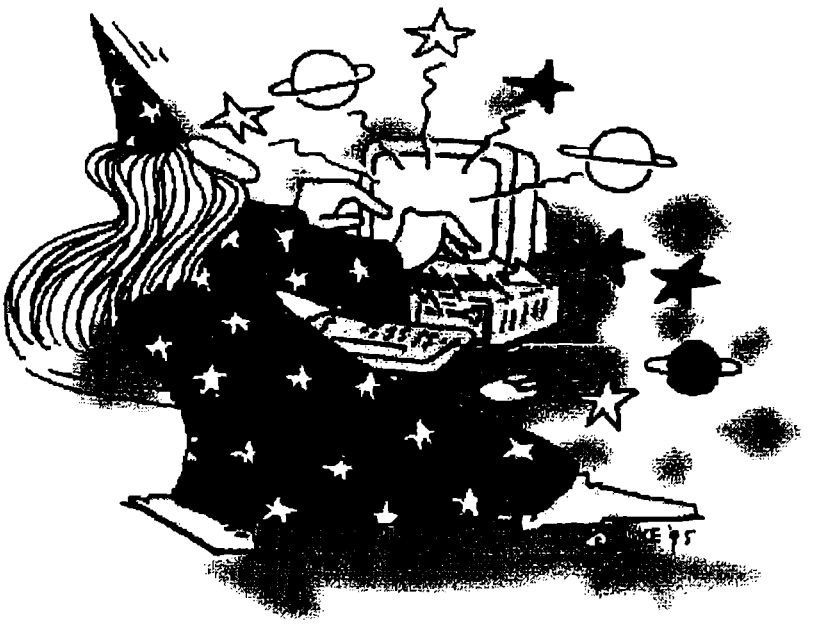
Acest procedeu se reia pentru fiecare din pătratele obţinute: rezultă `16` pătrate, fiecare având ataşat câte un cod format din două cifre ca mai jos

11	12	21	22
13	14	23	24
31	32	41	42
33	34	43	44

conform aceluiaşi procedeu fiecare pătrăţel se reîmparte, obţinându-se pătrăţele cu codurile formate din trei cifre ş.a.m.d. Problema cere:

a) dându-se trei numere întregi `n`, `l` şi `c` să se afişeze codul pătratului aflat pe linia `l` şi coloana `c` din matricea de ordinul `n` (codul pătrăţelului va fi format din `n` cifre).

b) invers, dându-se codul de `n` cifre al unui pătrat să se afle pe ce linie respectiv coloană se găseşte acesta.



6. Recursivitatea

6.1. Funcții recursive

1. Să se scrie o funcție recursivă care calculează valoarea |x| pentru un număr întreg c.

   Exemplu. |2|=2, |-2|=2.

2. Să se scrie un program care afișează valoarea A(n,x) pentru n și x citite de la tastatură unde $A \quad N^2 \rightarrow N$ este funcția lui Ackermann dată de ecuațiile:

$$A(0, x) = x + 1$$
$$A(n + 1,0) = A(n,1)$$
$$A(n + 1, x + 1) = A(n, A(n + 1, x))$$

   Exemplu. A(4,3)=253.

3. Se dă șirul definit astfel:

$$U_n = \begin{cases} 0 & \text{daca } n = 0 \\ 1 & \text{daca } n = 1 \\ U_{n-1} + U_{n-2} & \text{altfel} \end{cases}$$

   numit șirul lui Fibonacci. Se citește n, număr natural, să se calculeze și afișeze U_n.

   Exemplu. U_{20}=7675.

4. Se citește $x \in Z$. Se cere să se scrie un program pentru calculul funcției Manna-Punneli definită prin:

$$F(x) = \begin{cases} x - 1 & x \geq 12 \\ F(F(x + 2)) & x < 12 \end{cases}$$

   Exemplu. F(4)=11, F(20)=19.

5. Numărul de partiții ale unei mulțimi cu n elemente în k submulțimi este S(n,k) numit numărul lui Stirling de speța a doua, și este definit de relațiile:

$$S(n,1) = S(n, n) = 1$$
$$S(n + 1, k) = S(n, k - 1) + kS(n, k)$$

   Să se calculeze numărul partițiilor unei mulțimi cu n elemente în submulțimi de k elemente.

   Exemplu. S(5,3)=25.

6. Fiind date două numere naturale m, n să se afişeze valoarea P(m,n), unde P se defineşte prin:

$$P(x, y) = \begin{cases} 1, & \text{daca } y = 1 \text{ sau } y = x \\ 0, & \text{daca } y > x \\ P(x - y,1) + P(x - y,2) + \quad + P(x - y, y), & \text{altfel} \end{cases}$$

Exemplu. P(7,4)=3, P(35,7)=1367.

7. Să se determine şi să se afişeze valoarea f(n) pentru un n citit de la tastatură, funcţia f:N→N fiind definită prin relaţiile:

$$f(n) = n * n + f(n - 1)$$
$$f(1) = 0$$

Exemplu. f(10)=384.

8. Să se determine şi să se afişeze numărul întregilor n, 1<=n<=2002, pentru care f(n)=n, unde f:N→N este definită prin:

$$f(1) = 1$$
$$f(3) = 3$$
$$f(2k) = f(k)$$
$$f(4k + 1) = 2f(2k + 1) - f(k)$$
$$f(4k + 3) = 3f(2k + 1) - 2f(k)$$

Exemplu. Există 92 de valori ale lui n cu proprietatea enunţată.

9. Scrieţi un program pentru calculul recursiv al coeficienţilor binomiali după formula dată de triunghiul lui Pascal:

$$\binom{n}{k} = \begin{cases} \binom{n-1}{k-1} + \binom{n-1}{k} & \text{pentru } 0 < k < n \\ 1 & \text{altfel} \end{cases}$$

Exemplu. $\binom{7}{3} = 35$.

10. Scrieţi un program care afişează pe ecran valoarea k(B, n, T) unde k $N^3 \rightarrow N$ se defineşte prin:

$$k(B,1, T) = B^T$$
$$k(B, n,1) = B$$
$$k(B, n + 1, T + 1) = k(B, n, k(B, n + 1, T))$$

iar B, n, T se vor citi de la tastatură.

Exemplu. k(4,2,2)=4.

11. Scrieți o funcție sau o procedură recursivă care implementează următorul algoritm recursiv:

```
n#=A[n]*(n-1) #
```

și `0#=1`

Variabila tablou `A` (cu componente de tip întreg) este de asemenea pasat ca parametru procedurii.

Exemplu. Pentru `n=5` și vectorul `A=(21,3,74,6,4)` se obține valoarea funcției egală cu `111888`.

12. Scrieți o procedură sau funcție recursivă care implementează următorul algoritm recursiv:

```
n&=(n mod 7)+(n-1)&
```

și `0&=0`.

Exemplu. `15&=43`.

13. Să se scrie o funcție recursivă care calculează valoarea polinomului Laguerre, definit în felul următor:

$L_0(x) = 1$
$L_1(x) = 2 - x$
$nL_n(x) = (2n - x)L_{n-1}(x) - nL_{n-2}(x)$ pentrun > 1

Exemplu. $L_5(1.5) = -1.520833$.

14. Să se scrie o funcție recursivă care calculează valoarea polinomului Cebîșev, definit astfel:

$T_0(x) = 1$
$T_1(x) = 2x$
$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$ pentrun ≥ 2

Exemplu. $T_8(2.3) = 140661.628550$.

15. Scrieți un program care calculează valoarea polinomului lui Hermite definit recursiv astfel:

$H_0(x) = 1$
$H_1(x) = 2x$
$H_n(x) = 2xH_{n-1}(x) - 2(n - 1)H_{n-2}(x)$ pentru n > 1

Exemplu. $H_{10}(0.3) = 24755.2000$.

16. Să se scrie o funcție recursivă care determină suma elementelor dintr-un tablou de numere întregi.

Exemplu. Pentru `n=5` și vectorul `x=(12,9,45,3,22)` suma este `91`.

17. Scrieți o funcție recursivă care calculează $\sqrt{n}$. Se va citi de la tastatură n, a (o aproximație inițială a lui $\sqrt{n}$), și e (precizia pe care dorim să o obținem). Se va folosi relația:

$$RAD(n,a,e) = \begin{cases} a & \text{daca } |a^2 - n| < e \\ RAD\left(n, \dfrac{a^2 + n}{2a}, e\right), & \text{altfel} \end{cases}$$

Exemplu. Pentru n=5, a=1, e=0.0000001 => rad=2.23606797.

18. Să se scrie un program care calculează termenul al n-lea al șirului (x_n) definit astfel:

$$x_0 = 1 \quad x_1 = 1$$

$$x_{k+1} = x_k - \frac{x_k^3 - p}{x_k^2 + x_k x_{k-1} + x_{k-1}^2} \quad \text{pentru } k > 1$$

unde p este un număr real citit de la tastatură.

Exemplu. Pentru n=10, p=2.1 => x=1.280579165.

19. Scrieți un program care afișează primii 20 de termeni ai șirului t_n definit prin:

$$t_n - 3t_{n-1} - 4t_{n-2} = 0 \quad \text{pentru } n \geq 2$$
$$t_0 = 0, t_1 = 1$$

Exemplu. t_{18}=13107.

20. Fie șirul $a_0, a_1, \ldots, a_n, \ldots$ definit prin relațiile

$$a_0 = a$$

$$a_{n+1} = \begin{cases} \dfrac{a_n}{2} & \text{daca } a_n \text{ este par} \\ 3a_n + 1 & \text{altfel} \end{cases}$$

Pentru valoarea a citită de la tastatură să se afișeze indecșii i, i<=k dat de la tastatură, care satisfac condiția $a_i = 1$.

Exemplu. Pentru k=20, a=4 se vor afișa indicii 2,5,8,11,14,17,20.

21. Să se scrie un program care, folosind recursivitatea, calculează și afișează valoarea $z = a^b$, a, b întregi citiți de la tastatură, utilizând formula:

$$a^b = \begin{cases} a \cdot (a^{b-1}) & \text{daca } b > 0 \\ 1 & \text{daca } b = 0 \end{cases}$$

Exemplu. Pentru a=2 și b=5 rezultatul este 32.

22. Să se scrie câte o funcție recursivă pentru calculul următoarelor expresii:

a) $S=1+3+5+\ldots+(2n-1)$;

Exemplu. Pentru $n=10$ se obține $S=100$.

b) $P=1\cdot 4\cdot 7\cdot \ldots\cdot (3n-2)$;

Exemplu. Pentru $n=5$ obținem $P=3640$.

c) $P=2\cdot 4\cdot 6\cdot \ldots\cdot (2n)$;

Exemplu. Pentru $n=5$ se obține $P=3840$

d) $E=1\cdot 2-2\cdot 4+\ldots+(-1)^{n+1}n\cdot 2n$.

Exemplu. Pentru $n=8$ se obține $E=-72$.

23. Să se scrie un program care, folosind recursivitatea, calculează și afișează valoarea $z = a^b$, a, b întregi citiți de la tastatură, observând că:

$$a^b = \begin{cases} (a\ a)^{\lfloor b/2 \rfloor} & \text{daca b este par} \\ (a\ a)^{\lfloor b/2 \rfloor}\ a & \text{daca b este impar.} \\ 1 & \text{daca b = 0} \end{cases}$$

Exemplu. Pentru $a=11$ și $b=3$ se obține 1331.

24. Să se scrie un algoritm recursiv care calculează a^b folosind formula

$$a^b = \begin{cases} \left(a^{\frac{b}{2}}\right)^2 & \text{daca b este par} \\ a\left(a^{\frac{b-1}{2}}\right)^2 & \text{daca b este impar sib} \geq 1' \\ 1 & \text{daca b = 0} \end{cases}$$

întregii a și b sunt citiți de la tastatură.

Exemplu. $8^3=512$.

25. Să se scrie un program care calculează suma cifrelor unui număr natural citit de la tastatură. Se va folosi o funcție recursivă.

Exemplu. Pentru $n=2354$ obținem suma 14.

26. Să se scrie un program care, folosind recursivitatea, calculează produsul a b a două numere întregi utilizând doar operații de adunare și înmulțiri sau împărțiri la 2.

Indicație. Se utilizează formula:

$$a \quad b = \begin{cases} \left[\frac{a}{2}\right] \quad (2 \quad b) & \text{daca a este par} \\ \left[\frac{a}{2}\right] \quad (2 \quad b) + b & \text{daca a este impar} \\ 0 & \text{daca a = 0 sau b = 0} \end{cases}$$

Exemplu. 12*55=660 (=6·110=3·220=1·440+220=0·880+440+220).

27. Să se scrie un program care transformă un număr natural n, din baza 10 în baza b (1<b<10).

Exemplu. Pentru n=327, b=2 se obține 101000111.

28. Să se scrie o funcție recursivă care verifică dacă un număr întreg este prim sau nu.

Exemplu. Numărul n=1291 este prim.

29. Se introduce de la tastatură un vector x=(x[1],x[2],...,x[n]) cu n componente numere naturale cu cel mult 9 cifre. Să se afișeze suma

P(x[1])+P(x[2])+...+P(x[n])

unde P(a) este prima cifră a numărului a. Se va folosi o funcție recursivă care returnează prima cifră a unui număr natural de tip longint.

Exemplu. Pentru n=3 și vectorul x=(538,16,94) suma este 15.

30. Să se scrie un program care determină cel mai mare divizor comun a două numere întregi x și y utilizând proprietățile:

$$\text{c.m.m.d.c.}(x, y) = \begin{cases} \text{c.m.m.d.c.}(x - y, y), & \text{daca } x > y \\ \text{c.m.m.d.c.}(x, y - x), & \text{daca } y > x. \\ x & \text{daca } x = y \end{cases}$$

Exemplu. cmmdc(13455,12435)=15.

31. Să se scrie un program care calculează cel mai mare divizor comun a două numere folosind o funcție recursivă ce implementează algoritmul lui Euclid.

Exemplu. cmmdc(17182,24684)=242.

32. Se dă un număr natural n. Să se determine dacă n este un palindrom (egal cu răsturnatul său), utilizând recursivitatea.

Exemplu. 12321, 5, 22, sunt numere palidroame.

33. Să se scrie un program recursiv care testează egalitatea a două şiruri de caractere.

Exemplu. `'abc'='abc'`, `'abd'<>'abc'`, `'abcd'<>'ab'`

34. Fie `x`, `y` două şiruri de caractere. Scrieţi un program recursiv care să testeze dacă şirul `y` este o anagramă a şirului `x`.

Exemplu. Şirurile 'abcd' şi 'dabc' sunt anagrame.

35. Să se scrie un program care calculează suma: $x[1]^{y[1]}+x[2]^{y[2]}+...+x[n]^{y[n]}$ pentru `x`, `y` doi vectori de numere întregi citite de la tastatură. Se vor folosi două funcţii recursive, una pentru calculul puterii a^b, iar alta pentru calcularea sumei componentelor unui vector.

Exemplu. Pentru `n=3`, `x=(5,7,8)`, `y=(3,2,4)` obţinem suma `4270`.

36. Să se scrie o funcţie recursivă pentru a calcula produsul a două numere întregi prin adunări repetate.

Exemplu. `7*35=245`.

37. Să se scrie o funcţie recursivă pentru a împărţii două numere întregi folosind scăderi repetate.

Exemplu. `4587:65=70` rest `37`.

38. O persoană are de coborât `n` trepte. La fiecare pas el poate coborî `1` sau `2` trepte. Să se determine câte posibilităţi are persoana de a coborî scara.

Exemplu. Pentru `n=12` există `233` de variante de a urca scările.

39. Se dă un vector cu `n` numere întregi. Să se determine, folosind o funcţie recursivă maximul elementelor din vector.

Exemplu. Pentru `n=5` şi vectorul `x=(586,31,919,3,18)` maximul este `919`.

40. Se dă de la tastatură un vector `x` cu `n` componente numere naturale cu cel mult `9` cifre. Să se afişeze componentele care au suma cifrelor maximă. Se va folosi o funcţie recursivă pentru calculul sumei cifrelor unui număr.

Exemplu. Pentru `n=5` şi vectorul `x=(586,31,919,3,18)` suma maxima este `19`, iar elementele cu această sumă a cifrelor sunt `586` şi `919`.

41. Se dă un vector cu `n` numere întregi. Să se verifice, folosind o funcţie recursivă, dacă există măcar un element pozitiv în vector.

42. Mediana unui număr impar de numere este definită ca fiind numărul care are proprietatea că numărul de numere mai mari decât el este egal cu numărul de numere mai mici decât el. Să se scrie o funcţie recursivă care determină această mediană.

Exemplu. Mediana următoarelor numere `(2,23,7,3,5,1,98,6,10)` este `7`.

43. Se dă un tablou cu `n` numere întregi. Să se determine, folosind funcții recursive:
    a) numărul componentelor negative din vector;
    b) produsul elementelor pare din vector;
    c) produsul elementelor de pe pozițiile impare din vector;
    d) suma elementelor pare de pe pozițiile divizibile cu `k` din vector;
    e) suma pătratelor elementelor ce au cubul mai mic decât `k`;
    f) suma numerelor pozitive și prime din vector;
    g) numărul componentelor impare ale vectorului care aparțin intervalului `[a,b]`;
    h) de câte ori o componentă a vectorului este media aritmetică a componentelor vectorului.

    Exemplu. Pentru `n=8, k=4, a=3, b=8` și `x=(5,8,4,2,-6,15,-7,11)` se obține: a) 2; b) −384; c) 840; d) 2; e) 85; f) 7; g) 1; h) 1.

44. Se dau doi vectori `x` și `y` cu `n` componente fiecare. Folosind o funcție recursivă să se calculeze produsul scalar al celor doi vectori definit astfel:

$$(x, y) = \sum_{i=1}^{n} x_i y_i .$$

    Exemplu. Dacă `n=5` și `x=(5,8,4,11,3)` iar `y=(6,2,-5,4,9)` produsul scalar va fi 97.

45. Fie șirul `1,1,2,2,3,3,...,n,n,...`. Găsiți o definiție recursivă a acestui șir și scrieți apoi un program o implementează.

46. Fie ecuația de gradul II $x^2 - sx + p = 0$ cu $s,p \in R$ și $n \in N$. Fără a rezolva ecuația să se calculeze $S_n = x_1^n + x_2^n$.

    Indicație Se va căuta o formulă recursivă pentru S_n plecând de la observația că: $x_1^2 - sx_1 + p = 0$ și $x_2^2 - sx_2 + p = 0$.

    Exemplu. Pentru `s=2, p=7, n=10` suma este 24950.

47. Se consideră că diagonalele împart matricea pătratică `A` în 4 zone: nord, sud, est și vest (elementele de pe diagonală nu fac parte din nici o zonă). Să se scrie câte o funcție recursivă pentru a calcula:
    a) suma elementelor din nord din matricea `A`.
    b) produsul elementelor din vest din matricea `A`.

    Exemplu. Pentru `n=4` și matricea $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$ se obține: a) 5; b) 96.

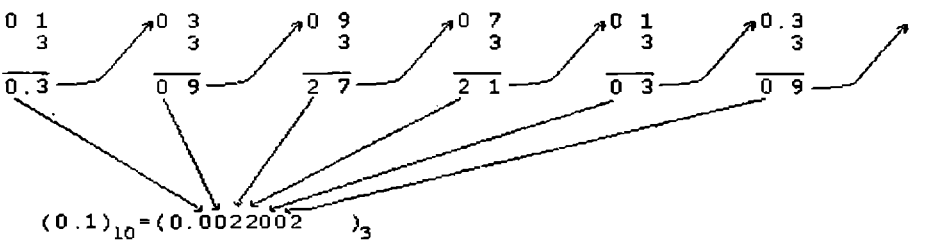
48. Pentru un număr natural `n`, citit de la tastatură, să se afişeze valoarea `f(n)`, unde `f:N→N` este definită prin `f(0)=f(1)=1` şi prin relaţiile:

$$f(2k) = 1 + \sum_{i=1}^{k} f(2i-1)$$

$$f(2k+1) = \sum_{i=0}^{k} f(2i)$$

Exemplu. `f(5)=8`; `f(10)=89`.

49. Convertirea unei fracţii zecimale subunitare într-o altă bază necesită ca fracţia să fie înmulţită repetat cu baza. Cifrele care apar la stânga punctului zecimal devin cifrele răspunsului în ordine de la stânga la dreapta. De exemplu, `0.1` este convertit la baza 3 după cum urmează:



Scrieţi o funcţie recursivă care converteşte o fracţie zecimală subunitară la o bază specificată. Funcţia se va numi `CONVERT` şi va avea 3 parametri, `f`, `b` şi `n`. `f` este o variabilă reală conţinând fracţia de convertit, `b` un întreg conţinând baza la care se face conversia (`b<=10`), `n` numărul de cifre cerute.

6.2. Proceduri recursive

50. Să se genereze primii `n` termeni ai unei progresii geometrice dacă se cunosc primul termen şi raţia.

Exemplu. Dacă primul termen este 3 iar raţia este 2 atunci primii 8 termeni ai progresiei geometrice sunt: 3,6,12,24,48,96,192,384.

51. Să se scrie un program care afişează în ordine inversă un şir de caractere de lungime arbitrară folosind recursivitatea. Introducerea se termină prin spaţiu.

Exemplu. Dacă se introduce şirul '`ABCD `' se va afişa '`DCBA`'.

52. Se dă un tablou cu `n` numere întregi. Să se scrie o procedură recursivă care afişează componentele vectorului.

53. Se citeşte de la tastatură, caracter cu caracter, un şir de caractere. Citirea se încheie la întâlnirea caracterului '$'. Folosind un algoritm recursiv, să se afişeze în ordinea inversă citirii, cifrele care apar în şir.

Exemplu. Dacă şirul introdus este 'AB78U6734HJD$' atunci se va afişa '437687'.

54. Scrieţi o procedură recursivă TRIM care elimină toate spaţiile dinaintea şi de după un text dat. De exemplu TRIM(' CMPT 81 ') returnează şirul 'CMPT 81'.

55. Să se scrie un program care descompune un număr dat n ca sumă de puteri distincte ale lui 2.

Indicaţie. Se va folosi o procedură care determină cea mai mare valoare p cu proprietatea că $2^p \le n$.

Exemplu. Pentru n=4978 se obţine descompunerea $2^1+2^4+2^5+2^6+2^8+2^9+2^{12}$.

56. Scrieţi o procedură recursivă care să implementeze următorul algoritm recursiv.

> Dacă suma lui x şi y este impară, afişează x şi y şi apelează recursiv algoritmul pentru x-1 şi y-1.
>
> Dacă suma lui x şi y este pară, apelează recursiv algoritmul pentru x-1 şi y.
>
> Dacă y este zero nu se face nimic.

unde x şi y sunt numere naturale nenule.

Exemplu. Pentru x=10 şi y=4 se va afişa:

```
9 4
8 3
7 2
6 1
```

57. Să se completeze în spirală, folosind o procedură recursivă, o matrice cu n linii şi n coloane cu primele n^2 numere întregi cu cel puţin 2 cifre, având cu proprietatea că suma cifrelor lor este divizibilă cu 5.

Exemplu. Pentru n=4 se obţine matricea $\begin{pmatrix} 14 & 19 & 23 & 28 \\ 69 & 73 & 78 & 32 \\ 64 & 87 & 82 & 37 \\ 55 & 50 & 46 & 41 \end{pmatrix}$.

58. Scrieţi o procedură recursivă care descompune un număr întreg dat în factori primi.

Exemplu. $19344=2^4*3*13*31$.

59. Să se scrie un program recursiv pentru determinarea şi afişarea valorilor din triunghiul lui Pascal până la nivelul k. Amintim că triunghiul lui Pascal este format din coeficienţii binomului lui Newton $(a+b)^n$ pentru n=1,2,...,k.

Exemplu. Pentru k=5 triunghiul este

```
    1
   1 1
  1 2 1
 1 3 3 1
1 4 6 4 1
```

60. Scrieţi un program recursiv care să implementeze următorul algoritm. Umple un tablou de caractere cu o serie de pătrate concentrice de x-uri. Fiecare pătrat trebuie să aibă latura cu patru caractere mai mică decât pătratul în care va fi încadrat (vezi figura). Procedura recursivă folosită primeşte ca parametru poziţia primei şi ultimei coloane a celui mai mare pătrat. Nici un pătrat nu va avea latura mai mică de 4.

```
x x x x x x x x x x x x x
x                       x
x   x x x x x x x x x   x
x   x                   x   x
x   x                   x   x
x   x   x x x x         x   x
x   x   x       x       x   x
x   x   x       x       x   x
x   x   x x x x         x   x
x   x                   x   x
x   x                   x   x
x   x x x x x x x x x   x
x                       x
x x x x x x x x x x x x x
```

61. Pentru un întreg n dat, să se construiască recursiv triunghiul de numere de mai jos fără a folosi nici o instrucţiune repetitivă:

```
n  n-1  n-2          3 2 1
n  n-1  n-2          3 2


n  n-1
n
```

62. Să se genereze toate permutările mulţimii {1,2,..,n}. Se va folosi următoarea observaţie. Permutările de ordin n se pot obţine din permutările de ordin n+1 punând valoarea n pe toate poziţiile posibile în această permutare. De exemplu din permutarea (2,1,3) de ordin 3 se obţin permutările (2,1,3,4), (2,1,4,3), (2,4,1,3), (4,2,1,3) de ordin 4.

63. Să se genereze toate elementele produsului cartezian

$$\underbrace{\{1,2,\ldots,n\} \times \{1,2,\ldots,n\} \times \ldots \times \{1,2,\ldots,n\}}_{\text{de } n \text{ ori}}.$$

64. Codul Fibonacci de ordinul n, $n \geq 2$, este secvența c_n. Dacă λ este șirul nul atunci codul Fibonacci se poate obține recursiv astfel:

$$c_2 = (\lambda) \qquad c_4 = (00,01,10)$$
$$c_3 = (0,1)$$

iar c_n, $n \geq 5$ se poate obține prin concatenarea a două subsecvențe: prima subsecvență se obține prefixând cu 0 fiecare codificare din c_{n-1}; a doua subsecvență se obține prefixând cu 10 fiecare codificare din c_{n-2}.

Exemplu $c_5 = (000,001,010,100,101)$

Elaborați un algoritm care construiește codul Fibbonacci pentru orice n dat.

65. Se știe că din numărul 4 se poate obține orice număr natural N scris în baza 10 prin aplicarea următoarelor operații:

a) se scrie la sfârșit cifra 4

b) se adaugă la sfârșit cifra 0

c) se împarte la 2 (dacă numărul este par).

Se cere să se scrie un program care produce un șir de numere construit conform regulilor precedente, șir în care primul număr este 4 iar ultimul este N. Nu se va folosi nici un tip de date structurat (array, string etc.)

Exemplu $4 \rightarrow 2 \rightarrow 1 \rightarrow 14 \rightarrow 140 \rightarrow 1404 \rightarrow 702 \rightarrow 7024$

Indicație. Se va folosi o procedură recursivă, iar șirul se va genera pornind invers de la N la 4, aplicând transformările inverse. (de exemplu transformarea de la punctul a devine: se elimină cifra 4 de la sfârșitul numărului)

66. Să se genereze seria Farey de ordinul n ($n \geq 1$) definită ca fiind seria conținând șirul crescător al fracțiilor reduse cuprinse între 0 și 1 care au numitorii mai mici sau egali cu n. De exemplu seria Farey de ordinul 7 este

$$\frac{0}{1},\frac{1}{7},\frac{1}{6},\frac{1}{5},\frac{1}{4},\frac{2}{7},\frac{1}{3},\frac{2}{5},\frac{3}{7},\frac{1}{2},\frac{4}{7},\frac{3}{5},\frac{2}{3},\frac{5}{7},\frac{3}{4},\frac{4}{5},\frac{5}{6},\frac{6}{7},\frac{1}{1}.$$

Se va folosi observația următoare: ea conține fracțiile 0/1 și 1/1 și în plus între fracțiile $\frac{a}{b}$ și $\frac{a'}{b'}$ ale seriei se află și seria $\frac{a+a'}{b+b'}$ dacă $b + b' \leq n$.

Să se afișeze și numărul de fracții din serie.

67. Fie şirul 1,2,3,4,5,10,20,40,... ai cărui termeni, începând cu al şaselea, satisfac relaţia $a_i=2a_{i-1}$. Să se scrie un algoritm recursiv care descompune un număr pozitiv n ca sumă de numere distincte din şirul definit mai sus.

Exemplu. Pentru n=4578 se obţine descompunerea

$$n=3+5+10+80+640+1280+2560.$$

68. Să se scrie un program care generează seria Farey de ordinul n (definită la problema anterioară) folosind următoarele observaţii în legătură cu aceste fracţii. Convenim să notăm seria Farey de ordinul n prin: $(x_1,y_1),(x_2,y_2),\ldots$ Se poate observa că au loc următoarele relaţii:

$$x_0 = 0, x_1 = y_0 = 1, y_1 = n$$

$$x_{k+2} = \left[\frac{y_k + n}{y_{k+1}}\right] x_{k+1} - x_k .$$

$$y_{k+2} = \left[\frac{y_k + n}{y_{k+1}}\right] y_{k+1} - y_k$$

69. Se citesc de la tastatură n cuvinte. Folosindu-se proceduri recursive să se elaboreze un program care afişează pe ecran o listă de cuvinte obţinută astfel:

    fiecărui cuvânt i se vor inversa literele;

    fiecare cuvânt astfel inversat va fi afişat în ordinea inversă citirii lui de la tastatură.

Nu se va folosi nici un vector sau altă structură auxiliară.

Exemplu. pentru n=3 şi cuvintele **mama**, **apoi**, **când** se va afişa: **dnac**, **iopa**, **amam**.

70. Fie polinomul P_n de grad n>=3, dat prin:

$$P_0 = 1, \quad P_1 = x,$$

$$P_{k+1} = 2*x*P_k - P_{k-1}, \quad 2<=k<=n.$$

Să se scrie un program care determină coeficienţii polinomului dat.

Exemplu. Polinomul de grad 5 obţinut cu formulele de recurenţă de mai sus este

$$P(x)=16x^5-20x^3+5x.$$

71. Să se scrie o procedură recursivă care generează vectorul p_n, cu 2^n-1 componente definit astfel:

$$p_1 = (1)$$
$$p_{k+1} = (p_k, k + 1, p_k)$$

Exemplu. p_4=(1,2,1,3,1,2,1,4,1,2,1,3,1,2,1).

72. Într-un oraş se stabilesc 3n obiective economice. Pentru construcţia lor se lansează o licitaţie la care orice antreprenor poate câştiga 3 obiective. Câştigătorul licitaţiei are dreptul să-şi aleagă cele trei obiective O_1, O_2, O_3. El face acest lucru astfel încât suma distanţelor dintre cele trei obiective să fie minimă şi să nu depăşească o valoare fixată de k kilometri.

    După ce s-au adjudecat primele trei obiective, licitaţia se reia cu obiectivele rămase. Procesul se încheie dacă s-au terminat obiectivele sau nu se mai pot stabili triplete O_1, O_2, O_3 în condiţiile date. Se cere să se listeze rezultatele licitaţiilor succesive, afişând de fiecare dată şi suma distanţelor. Se va folosi o procedură recursivă.

73. Să se scrie un program care citeşte un număr natural n (1<n<16) şi scrie pe ecran un şir care conţine numerele între 1 şi 2^n, şir creat astfel:

    se porneşte cu şirul iniţial format din 1 şi 2;

    exact la mijlocul acestuia se inserează secvenţa formată de 3 şi 4 obţinându-se şirul 1,3,4,2;

    exact în mijlocul şirului astfel format se inserează secvenţa formată de numerele 5,6,7,8 obţinându-se şirul 1,3,5,6,7, 8,4,2 etc;

    În general: la mijlocul şirului format de primele 2^i numere naturale se inserează secvenţa ordonată formată din următoarele 2^i numere naturale.

74. Să se scrie un program care folosind recursivitatea roteşte cu 180° un tablou pătratic.

    Exemplu. Matricea $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$ devine $\begin{pmatrix} 16 & 15 & 14 & 13 \\ 12 & 11 & 10 & 9 \\ 8 & 7 & 6 & 5 \\ 4 & 3 & 2 & 1 \end{pmatrix}$.

75. Scrieţi o variantă recursivă a algoritmului de sortare prin inserţie.

    Indicaţie. Pentru a sorta T[1..n], sortăm recursiv T[1..n-1] şi inserăm T[n] în tabloul sortat T[1..n-1].

76. Fie n un număr natural pozitiv. Să se genereze toate succesiunile de n (n<20) caractere '*' şi '-'

    Exemplu. Pentru n=3 şirurile sunt:

---	*--
--*	*-*
-*-	**-
-**	***

77. Scrieți un program care folosind recursivitatea inversează secțiunea `b[i j]` a vectorului `b`. Adică, dacă inițial `b[i j]` = $(b_i, b_{i+1}, ..., b_j)$, după terminare `b[i j]` = $(b_j, ..., b_{i+1}, b_i)$. Indicii `i` și `j` se vor cere de la tastatură și se va verifica să fie în limitele vectorului `(1:n)` și $i \leq j - 1$.

Exemplu. Pentru `n=8, b=(1,2,3,4,5,6,7,8), i=2, j=5` se obține `b=(1, 5,4,3,2,6,7,8)`.

78. Să se scrie un program care implementează următorul algoritm recursiv (vezi figura de mai jos):

    în mijlocul ecranului se desenează un pătrat cu latura de `L` (dată de la tastatură);

    în fiecare din vârfurile pătratului se desenează câte un pătrat cu latura egală cu `L/2`, după care se repetă pasul al doilea pentru fiecare din aceste patru pătrate.

    Algoritmul se termină:

    a) când s-a desenat nivelul `n` (`n` dat de la tastatură)

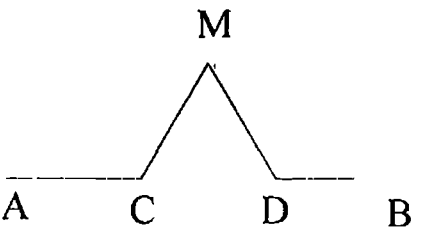
    b) latura pătratelor desenate este mai mică decât o valoare `L1` dată de la tastatură.

79. *Triunghiul lui Sierpinski.* Se dă un triunghi oarecare, prin coordonatele vârfurilor triunghiului. Se împarte triunghiul în patru triunghiuri de arie egală ducând cele trei linii mijlocii ale triunghiului. Se repetă construcția pentru triunghiurile mici formate, cu excepția triunghiului format doar din linii mijlocii (cel mai interior din cele patru). Se repetă procedeul de o infinitate de ori. Problema pusă de Sierpinski este de a determina aria suprafeței rămase după eliminarea triunghiurilor construite la fiecare pas numai din linii mijlocii. Ceea ce cere problema noastră nu este însă să determinați această arie, ci să ilustrați problema lui Sierpinski prin aplicarea de `n` ori a procedeului.

    Indicație. Triunghiurile a căror suprafață se elimină se vor haşura folosind o haşură SolidFill (de exemplu).

80. Să se deseneze un pătrat cu diagonalele paralele cu axele de coordonate. Să se deseneze pătratul obținut prin unirea mijloacelor laturilor pătratului inițial. Să se repete procedeul pe `n` ori cu fiecare pătrat nou obținut. Fiecare pătrat va fi haşurat cu o culoare generată aleator.

81. *Linia de coastă Koch.* Fie un segment de dreaptă AB, dat prin coordonatele capetelor, şi punctele C şi D care împart segmentul în trei părţi egale. Construiţi un triunghi echilateral cu latura CD al cărui vârf se notează cu M (vezi figura). Se şterge segmentul CD şi se procedează cu segmentele AC, CM, MD şi DB exact la fel cum s-a procedat cu AB. Se repetă procedeul de n ori.



82. Să se deseneze curba lui Kock pentru toate cele trei laturi ale unui triunghi echilateral, obţinându-se *curba fulgului de zăpadă.*

83. Să se deseneze un segment de dreaptă de o anumită lungime dată de la tastatură, paralel cu axa Ox. Se şterge porţiunea din mijlocul segmentului de o lungime egală cu a cincea parte din lungimea segmentului. Se repetă procedeul de n ori cu fiecare din segmentele rămase. Figura rămasă se numeşte *praful lui Cantor.*

84. Se desenează un pătrat cu latura de o anumită lungime. Se împarte pătratul în 9 pătrate egale şi se haşurează pătratul din mijloc. Se repetă de n ori cu fiecare din cele 8 pătrate nehăsurate de la fiecare pas. Figura obţinută se numeşte *covorul lui Sierpiński.*

85. Se desenează un pătrat cu latura de o anumită lungime. În exteriorul lui se desenează alte pătrate cu latura egală cu jumătate din latura pătratului iniţial ca în figura de mai jos. Se repetă procedeul de n ori, cu pătratele nou obţinute la fiecare pas. Figura obţinută se numeşte *scara diavolului.*

6.3. Recursivitatea indirectă

86. Să se determine valoarea funcției

$$f(x, y) = \begin{cases} x & \text{pentru } x \geq y \\ g(x, y - 1) & \text{pentru } x < y \end{cases}$$

pentru x și y întregi citiți de la tastatură unde:

$$g(x, y) = \begin{cases} x + y - 5 & \text{pentru } x + y \geq 5 \\ 2 \quad f(0, y) & \text{pentru } x + y < 5 \end{cases}.$$

Exemplu. f(5,20)=19, f(1,2)=0, f(1,4)=0, f(2,5)=1.

87. Pentru un număr natural n, citit de la tastatură, să se afișeze valorile f(n) și g(n), unde f,g:N→N sunt definite prin

f(0)=f(1)=1

g(0)=2, g(1)=1

g(n+2)=5f(n)-g(n) pentru n>=0

f(n+1)=g(n+1)-f(n-1) pentru n>=1.

Exemplu. f(5)=8, g(5)=11: f(4)=5, g(4)=7.

88. De la tastatură se introduc 3 numere reale a, b, c și un număr natural n. Să se afișeze valorile f(n), g(n) și h(n) unde f,g,h :N → R și

$$f(0) = a \qquad g(0) = b \qquad h(0) = c$$

$$f(k + 1) = \frac{g(k) + h(k)}{2}$$

$$g(k + 1) = \frac{f(k) + h(k)}{2} \qquad \text{pt. } k \geq 0.$$

$$h(k + 1) = \frac{f(k) + g(k)}{2}$$

Se vor folosi funcții recursive.

Exemplu. Dacă a=1, b=2.5, c=4, atunci

f(5)=2.546875.

g(5)=2.5,

h(5)=2.453125.

7. Metode de programare

7.1. Backtracking obişnuit

1. Să se genereze toate permutările mulţimii {1,2,...,n}.

   Exemplu. Pentru n=3 se vor afişa permutările (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), (3,2,1).

2. Să se genereze toate permutările mulţimii {1,2,...,n} cu proprietatea că diferenţa în modul dintre oricare două numere consecutive este cel puţin egală cu o valoare v citită de la tastatură.

   Exemplu. Pentru n=4 şi v=2 singurele soluţii sunt (2,4,1,3) şi (3,1,4,2).

3. Să se genereze toate permutările lexicografice ale unui cuvânt citit de la tastatură.

   Exemplu. Permutările cuvântului 'casa' sunt: 'casa', 'caas', 'csaa', 'acsa', 'acas', 'asca', 'asac', 'aacs', 'aasc', 'scaa', 'saca', 'saac'

4. Să se genereze toate permutările mulţimii $\{a_1, a_2, ..., a_n\}$ de numere întregi citite de la tastatură.

   Exemplu. Pentru n=3 şi mulţimea {2,5,9} se afişează (2,5,9), (2,9,5), (5,2,9), (5,9,2), (9,2,5), (9,5,2).

5. Se citeşte de la tastatură un număr natural n (n<15). Să se genereze toate şirurile formate din numerele 1,2,...,n astfel încât fiecare valoare să apară exact o dată într+un şir, iar valorile pare să se găsească totdeauna la locul lor: 2 pe a doua poziţie din şir, 4 pe a patra poziţie din şir etc. Fiecare şir se va afişa pe un rând al ecranului, cu spaţii între elementele ce-l formează.

   Exemplu. Pentru n=6, se vor afişa pe ecran (nu neapărat în această ordine):

   1 2 3 4 5 6

   1 2 5 4 3 6

   3 2 1 4 5 6

   3 2 5 4 1 6

   5 2 1 4 3 6

   5 2 3 4 1 6

6. Să se genereze toate cele A_n^k aranjamente de p elemente ale mulțimii {1,2,...,n}.

Exemplu. Pentru n=4 și k=3 se va afișa: (1,2,3), (1,2,4), (1,3,2), (1,3,4), (1,4,2), (1,4,3), (2,1,3), (2,1,4), (2,3,1), (2,3,4), (2,4,1), (2,4,3), (3,1,2), (3,1,4), (3,2,1), (3,2,4), (3,4,1), (3,4,2), (4,1,2), (4,1,3), (4,2,1), (4,2,3), (4,3,1), (4,3,2).

7. Să se genereze toate partițiile mulțimii {1,2,...,n}.

Exemplu. Pentru n=3 se obțin următoarele partiții:

{1,2,3}

{1,2} {3}

{1,3} {2}

{1} {2,3}

{1} {2} {3}.

8. Fie $a=(a_1,a_2,...,a_n)$, $n\geq 3$, un șir de numere reale. Să se obțină din a prin permutări tabloul $b=(b_1,b_2,...,b_n)$ care să îndeplinească următoarele condiții:

b(i)>=b(i+1), pentru i=1, 2, ..., k-1;

b(i)<=b(i+1), pentru i=k, ..., n-1;

unde k este n/2 pentru n par și (n+1)/2 pentru n impar.

Exemplu. Pentru vectorul a=(1,2,3,4,5) se obțin soluțiile (3,2,1,4,5), (4,2,1,3,5), (4,3,1,2,5), (5,2,1,3,4), (5,3,1,2,4), (5,4,1,2,3).

9. Să se genereze toate partițiile mulțimii $\{a_1,a_2,...,a_n\}$.

Exemplu. Pentru n=3 se obțin următoarele partiții ale mulțimii {2,5,9}:

{2,5,9}

{2,5} {9}

{2,9} {5}

{2} {5,9}

{2} {5} {9}.

10. Să se genereze produsul cartezian a n mulțimi $A_1,A_2,...,A_n$, cu $A_i=\{1,2,...,a_i\}$.

Exemplu. Pentru n=3 și $a_1=2$, $a_2=3$, $a_3=2$, se obțin elementele (1,1,1), (1,1,2), (1,2,1), (1,2,2), (1,3,1), (1,3,2), (2,1,1), (2,1,2), (2,2,1), (2,2,2), (2,3,1), (2,3,2).

11. Să se genereze cele C_n^k combinări ale mulțimii {1,2,...,n}.

Exemplu. Pentru n=4 și k=3 se obțin combinările (1,2,3), (1,2,4), (1,3,4), (2,3,4).

12. Să se genereze toate submulțimile nevide ale mulțimii $A=(a_1,a_2,a_3,...,a_n)$.

Exemplu. Submulțimile mulțimii {2,5,9,10} sunt {2}, {5}, {9}, {10}, {2,5}, {2,9}, {2,10}, {5,9}, {5,10}, {9,10}, {2,5,9}, {2,5,10}, {2,9,10}, {2,5,9,10}.

13. Să se genereze toate funcțiile injective f:{1,2,...,n} → {1,2,...,m} (m>=n).

Exemplu. Pentru n=2 și m=3 se obțin funcțiile:

f(1)=1	f(2)=2	f(1)=2	f(2)=3
f(1)=1	f(2)=3	f(1)=3	f(2)=1
f(1)=2	f(2)=1	f(1)=3	f(2)=2

14. Se citește un număr natural n. Să se determine numărul maxim obținut prin eliminarea din numărul dat a k cifre.

Exemplu. Pentru n=1324 și k=2 se obține valoarea 34.

15. Să se genereze toate funcțiile surjective f:{1,2,...,n} → {1,2,...,m}.

Exemplu. Pentru n=3 și m=2 se obțin funcțiile

f(1)=1	f(2)=1	f(3)=2	f(1)=2	f(2)=1	f(3)=1
f(1)=1	f(2)=2	f(3)=1	f(1)=2	f(2)=1	f(3)=2
f(1)=1	f(2)=2	f(3)=2	f(1)=2	f(2)=2	f(3)=1

16. Să se genereze toate numerele naturale de n cifre care sunt supermultiple de k adică numărul respectiv și toate numerele ce se obțin din el prin eliminarea cifrelor sale începând cu cifra unităților sunt multiple de k.

Exemplu. De exemplu 246 este supermultiplu de 2 pentru că 246, 24 și 2 sunt multiple de 2.

17. Să se afișeze toate numerele naturale de n cifre care conțin k cifre de 1.

Exemplu. Pentru n=3 și k=2 se obțin soluțiile 101, 110, 112, 113, ..., 119, 121, 131, 141, ...

18. Se citește de la tastatură un număr natural n. Să se genereze toate numerele întregi a căror reprezentare în baza 2 au același număr de cifre 0 (semnificative) și respectiv 1 ca și reprezentarea în baza 2 a numărului n.

Exemplu. Pentru n=53 se vor afișa numerele 39, 43, 45, 46, 51, 53, 54, 57, 58, 60.

19. Să se scrie un program care afişează toate numerele de n cifre egale cu de k ori produsul cifrelor.

Exemplu. Pentru n=2, k=6 singura soluţie este 12.

20. Fie M o mulţime de numere naturale strict pozitive şi un număr natural S din N^*. Să se afişeze toate submulţimile mulţimii M cu proprietatea că suma valorilor elementelor submulţimii este egală cu S.

Exemplu. Pentru mulţimea M={1,2,3,4,5} şi S=8 soluţiile sunt: (3,5), (1,3,4), (1,2,5).

21. Pe fiecare din cele n scaune dintr-o încăpere, aşezate unul lângă altul pe un rând, se găseşte câte un copil. Între oricare doi copii cu scaunele vecine se iscă neînţelegeri. Să se reaşeze copiii pe scaune astfel încât foştii vecini să fie despărţiţi de cel puţin doi copii.

Exemplu. Pentru n=6 singurele soluţii sunt (2,4,6,1,3,5), (5,3,1,6,4,2), fiecare număr indicând copilul aflat iniţial în poziţia respectivă.

22. Un copil doreşte să pună n bile numerotate cu numere între 1 şi n în m cutii, numerotate de la 1 la m. Să se afişeze toate modalităţile pe care le are copilul de a pune bilele în cutii ştiind că într-o cutie încap cel mult n bile, însă numărul de cutii este mai mic decât numărul de bile (m<n).

Exemplu. n=3 bile se pot distribui în m=2 cutii astfel:

```
cutia 1: 1, 2, 3        cutia 2: -
cutia 1: 1, 2           cutia 2: 3
cutia 1: 1, 3           cutia 2: 2
cutia 1: 1              cutia 2: 2, 3
cutia 1: 2, 3           cutia 2: 1
cutia 1: 2              cutia 2: 1, 3
cutia 1: 3              cutia 2: 1, 2
cutia 1: -              cutia 2: 1, 2, 3
```

23. Un copil doreşte să pună n bile numerotate cu numere între 1 şi n în n cutii, numerotate de la 1 la n. Să se afişeze toate modalităţile pe care le are copilul de a pune bilele în cutii ştiind că într-o cutie încap cel mult m bile, m<=n.

Exemplu. Pentru n=3 bile şi maxim m=2 bile într-o cutie se obţin următoarele soluţii:

```
cutia 1: 1 2     cutia 2: 3       cutia 3: -
cutia 1: 1 2     cutia 2: -       cutia 3: 3
cutia 1: 1 3     cutia 2: 2       cutia 3: -
cutia 1: 1       cutia 2: 2 3     cutia 3: -
cutia 1: 1       cutia 2: 2       cutia 3: 3
```

etc.

24. Dându-se n cutii şi m obiecte (m>n), să se tipărească toate posibilităţile de repartizare a obiectelor în cutii, precum şi numărul acestor posibilităţi, ştiind că:

în prima cutie se află un obiect;

în a doua cutie se află unul sau două obiecte;

în a treia cutie se află două sau trei obiecte;

în a k-a cutie se află k-1 sau k obiecte (1<k<n);

în a n-a cutie se află n-1 obiecte.

Exemplu. Pentru n=4 cutii şi m=9 obiecte problema are 5040 de soluţii una dintre acestea fiind:

```
Cutia 1: 1
Cutia 2: 2 5
Cutia 3: 3 4 9
Cutia 4: 6 7 8
```

Pentru n=4 cutii şi n=10 obiecte problema nu are nici o soluţie.

25. Se dau n bile albe şi albastre. Fiecare bilă are o etichetă de tip caracter. Să se afişeze toate posibilităţile de a selecta k bile din care cel puţin a să fie albe.

Exemplu. Pentru a=2, k=3, n=5 bile de având culorile (alb,albastru, albastru,alb,alb) (observaţie: pentru uşurinţă culorile se pot codifica 1 alb şi 2 albastru) şi etichetele (a,b,e,d,f) soluţiile sunt: (e,d,f), (b,d,f), (a,d,f), (a,e,f), (a,e,d), (a,b,f), (a,b,d).

26. Se dau n bile din care p sunt albe iar restul albastre. Fiecare bilă are o etichetă de tip caracter. Să se afişeze toate posibilităţile de a selecta k bile din care cel puţin a să fie albe, iar cel puţin o bilă albă să aibă ca etichetă o vocală.

Exemplu. Pentru a=2, k=3, n=5 bile de având culorile (alb,albastru, albastru,alb,alb) şi etichetele (a,b,e,d,f) soluţiile sunt: (a,d,f), (a,e,f), (a,e,d), (a,b,f), (a,b,d).

27. Fiind dată o matrice A (n×n) cu elemente numere naturale, se cere să se determine cea mai mare sumă a n valori, luate din linii şi coloane diferite.

Exemplu. Pentru n=3 şi matrice $A = \begin{pmatrix} 1 & 3 & 5 \\ 9 & 2 & 4 \\ 6 & 1 & 47 \end{pmatrix}$ suma maximă este 59 (=3+9+47).

28. Să se genereze toate matricele binare cu n linii şi n coloane astfel încât fiecare coloană a matricei să conţină exact o singură valoare de 0.

Exemplu. Pentru n=2 matricele obţinute sunt $\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, $\begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$.

29. Să se calculeze suma tuturor numerelor formate numai din cifre impare distincte.

Exemplu. Suma este 7367625 (=1+13+135+1357+13579+1359+13597+...+97531).

30. Se dau n persoane {1,2,...,n} şi p perechi de forma (i,j) cu semnificaţia că i este fiul lui j. Să se aşeze cele n persoane în rând astfel încât fiecare persoană să se afle înaintea fiilor săi.

Exemplu. Pentru n=10, p=9 şi perechile (4,1), (4,5), (4,6), (1,8), (6,3), (6,7), (8,10), (8,9), (8,2) o soluţie este (2,3,5,7,6,9,10,8,1,4), alta (2,5,9,10,8,3,7,1,6,4) etc.

31. Pentru un n dat, să se genereze toate şirurile de 2n+1 termeni nenegativi, cu proprietatea că $a_1=0$, $a_{2n+1}=0$ şi $|a_i-a_{i+1}|=1$, $i=1,2,...,2n$.

Datele de ieşire se vor genera în fişierul SIR.TXT, cu structura:

pe o linie numărul şirurilor generate;

pe celelalte linii elementele şirului separate prin spaţiu.

Exemplu. Pentru n=3 se obţin şirurile (0,1,0,1,0,1,0), (0,1,0,1,2,1,0), (0,1,2,1,0,1,0), (0,1,2,1,2,1,0), (0,1,2,3,2,1,0).

32. Să se genereze toţi vectorii $a=(a_1,a_2,...,a_n)$ cu următoarele proprietăţi:

$a_1=1$ sau 2

$a_n=8$ sau 9

$|a_i-a_{i-1}|=2$ sau 3.

Exemplu. Pentru n=4 se obţin următorii vectori (1,3,5,8), (1,3,6,8), (1,3,6,9), (1,4,6,8), (1,4,6,9), (1,4,7,9), (2,4,6,8), (2,4,6,9), (2,4,7,9), (2,5,7,9).

33. Scrieţi un program care afişează toate modalităţile de a aranja elementele unui vector dat de numere întregi, astfel încât în şirul rezultat să nu existe două elemente alăturate negative.

Exemplu. Pentru n=4 şi vectorul (5,-3,-7,2) se obţin următoarele soluţii: (5,-3,2,-7), (5,-7,2,-3), (-3,5,-7,2), (-3,5,2,-7), (-3,2,5,-7), (-3,2,-7,5), (-7,5,-3,2), (-7,5,2,-3), (-7,2,5,-3), (-7,2,-3,5), (2,-3,5,-7), (2,-7,5,-3)

34. Se dă un cuvânt de maxim 20 de litere. Să se afişeze toate cuvintele ce conţin literele distincte ale cuvântului iniţial şi nu conţin două vocale sau două consoane alăturate.

Exemplu. Pentru cuvântul `'apelat'` se vor afişa cuvintele: `'palet'`, `'patel'`, `'pelat'`, `'petal'`, `'lapet'`, `'latep'`, `'lepat'`, `'letap'`, `'tapel'`, `'talep'`, `'tepal'` `'telap'`

35. Să se afişeze toate numerele de n cifre (n<=10) pentru care suma cifrelor este egală cu 5n.

Exemplu. Pentru n=2 numerele generate sunt 19, 28, 37, 46, 55, 64, 73, 82, 91.

36. Să se genereze toate numerele formate din n cifre distincte cu proprietatea că suma cifrelor este S.

Exemplu. Pentru n=3 şi S=22 se obţin numerele 589, 598, 679, 697, 769, 796, 859, 895, 958, 967, 976, 985.

37. Să se genereze toate numerele de cel mult n cifre, formate doar din cifre pare, cifre aflate în ordine strict crescătoare.

Exemplu. Pentru n=4 se vor afişa numerele 2, 24, 246, 2468, 248, 26, 268, 28, 4, 46, 468, 48, 6, 68, 8

38. Să se genereze toate numerele naturale formate din n cifre, fiecare număr generat având următoarele proprietăţi:

conţine numai cifre din mulţimea $\{c_1, c_2, c_3, c_4\}$;

orice două cifre alăturate sunt fie ambele pare fie ambele impare.

Numărul natural n precum şi cifrele c_1, c_2, c_3, c_4 se citesc de la tastatură.

Exemplu. Pentru n=3 şi c1=1, c2=2, c3=5, c4=6 se obţin soluţiile: 111, 115, 151, 155, 222, 226, 262, 266, 511, 515, 551, 555, 622, 626, 662, 666.

39. Să se genereze toate numerele prime de n cifre (n<=10) formate numai cu ajutorul cifrelor c1, c2 şi c3 citite de la tastatură.

Exemplu. Pentru n=4 şi c=(1,5,3) soluţiile sunt 1151, 1153, 1511, 1553, 1531, 5113, 5153, 5531, 5351, 5333, 3511, 3533, 3313, 3331.

40. Să se genereze toate şirurile de lungime n, formate numai din literele `'A'` şi `'M'`, şiruri care să nu aibă două litere `'A'` alăturate. Valoarea lui n se citeşte de la tastatură (0<n<13). Fiecare şir se va afişa pe un rând al ecranului, fără spaţii între litere.

Exemplu. Pentru n=5 se vor afişa următoarele şiruri: `'AMAMM'`, `'AMMAM'`, `'AMMMA'`, `'MAMAM'`, `'MAMMA'`, `'MMAMA'`, `'AMAMA'`, `'AMMMM'`, `'MAMMM'`, `'MMAMM'`, `'MMMAM'`, `'MMMMA'`, `'MMMMM'`

41. Scrieţi un program care afişează pe ecran toate secvenţele de n litere din mulţimea {c1,c2,c3,...,cp}, secvenţe care se pot construi respectând următoarele reguli: nu plasăm două litere identice una după alta, şi trebuie să utilizăm exact [n/2] litere c1. Literele c1,...,cp precum şi valoarea lui n se citesc de la tastatură.

Exemplu. Pentru n=4, p=3 şi literele 'a', 'b', 'x' se vor obţine următoarele soluţii: 'abab', 'abxa', 'abax', 'axab', 'axax', 'axba', 'baba', 'baxa', 'xaba', 'xaxa'.

42. Se dau p litere mici ale alfabetului englez. Se cere să se genereze toate cuvintele de lungime n (n<=p) care nu conţin trei vocale sau trei consoane alăturate. O literă nu se poate repeta în cadrul cuvântului.

Exemplu. Pentru n=3, p=3 şi literele 'a', 'e', 'i', 'm', şirurile generate sunt: 'aem', 'aim', 'ame', 'ami', 'eam', 'eim', 'ema', 'emi', 'iam', 'iem', 'ima', 'ime', 'mae', 'mai', 'mea', 'mei', 'mia', 'mie'

43. Fiind date două numere a şi b a căror reprezentare binară ocupă 16 biţi, spunem că a este şeful lui b dacă pentru orice poziţie binară 0 a lui a poziţia corespunzătoare din b este 0. Se citeşte n, număr natural care se poate reprezenta pe 16 poziţii binare. Să se genereze toţi şefii numărului n.

Exemplu. 767 este şeful lui 123 ($767=0000001011111111_2$; $123=0000000001111011_2$), dar 1 nu este şeful lui 123 ($4=0000000000000001_2$ deci cifra a doua din 1 în baza 2 este 0 iar în 123 este 1 deci 123 nu poate fi subalternul lui 1).

44. Un club de fotbal trebuie să participe cu o echipă la un meci important din campionatul naţional. Cunoscând numărul de portari, atacanţi, apărători şi mijlocaşi pe care îi are clubul şi ştiind că o echipă are exact un portar şi cel puţin câte doi atacanţi, doi mijlocaşi şi doi apărători, iar numărul total de sportivi este de 11 (nu se au în vedere rezervele), să se afişeze toate posibilităţile de a forma o echipă.

Exemplu. Dacă jucătorii disponibili sunt: portari: A, C, L; atacanţi: B, F, G, K, M; mijlocaşi: D, E, J, N, O; respectiv apărători: H, I, P, Q; atunci două soluţii ale problemei sunt: (C,D,E,I,J,K,M,N,O,P,Q) şi (D,F,G,I,J,K,L,N,O,P,Q).

45. Fiind date n cuburi etichetate de la 1 la n, de laturi L_i şi culori C_i, 1<=i<=n, să se afişeze toate turnurile de k cuburi care se pot forma, astfel încât cuburile din turn să aibă laturile în ordine descrescătoare, iar culorile cuburilor alăturate să fie diferite.

Exemplu. Pentru n=5, k=3, lungimile laturilor L=(1,2,3,4,5) şi culorile cuburilor C=(1,2,1,3,4) atunci soluţiile sunt (3,2,1), (4,2,1), (4,3,2), (5,2,1), (5,3,2), (5,4,1), (5,4,2), (5,4,3).

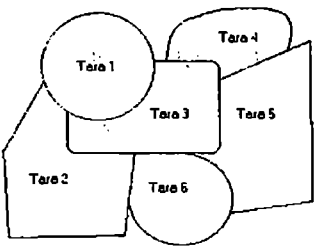
46. Se citesc denumirile a `n` culori (`string[16]`). Să se alcătuiască toate steagurile tricolore posibile astfel încât în mijloc să se afle doar una din ultimele două culori citite. Se va afişa pe fiecare rând:

numărul drapelului: culoare1, culoare2, culoare3.

Exemplu. Pentru `n=4` şi culorile `c=(roşu,albastru,galben,alb)`, steagurile sunt: (`roşu,galben,albastru`), (`roşu,galben,alb`), (`roşu, alb,albastru`), (`roşu,alb,galben`), (`albastru,galben,roşu`), (`albastru,galben,alb`), (`albastru,alb,roşu`), (`albastru,alb, galben`), (`galben,alb,roşu`), (`galben,alb,albastru`), (`alb,galben, roşu`).

47. Să se coloreze `m` ţări cu `n` culori astfel încât oricare două ţări vecine să fie colorate diferit. Pentru fiecare ţară se vor citi de la tastatură vecinii ei.

Exemplu. Pentru o hartă de forma celei de mai jos (`m=6`):



şi `n=3` se obţin următoarele soluţii: `(1,2,3,1,2,1)`, (adică ţara `1` are culoarea `1`, ţara `2` are culoarea `2`, ţara `3` culoarea `3`, ţara `4` culoarea `1`, ţara `5` culoarea `2`, ţara `6` culoarea `1`), `(1,3,2,1,3,1)`, `(2,1,3,2,1,2)`, `(2,3,1,2,3,2)`, `(3,1,2,3,1,3)`, `(3,2,1,3,2,3)`.

48. Să se determine numărul minim de culori necesare pentru a colora `m` ţări astfel încât oricare două ţări vecine să fie colorate diferit. Să se afişeze de asemenea o modalitate de colorare cu acest număr minim de culori. Pentru fiecare ţară se vor citi de la tastatură vecinii ei.

Exemplu. Pentru `m=6` şi harta de mai sus se obţine numărul minim de culori necesare este `3` şi oricare din soluţiile de la problema anterioară poate fi considerată soluţie pentru această problemă.

49. Se dau `n` numere întregi $x_1, x_2, \ldots, x_n$. Să se determine o aranjare a acestor numere pe un cerc astfel încât suma produselor de câte două numere vecine să fie maximă.

Exemplu. Pentru `N=6`, `x=(1,2,3,4,5,6)` şi aranjarea `2-5-1-6-4-3-2` suma este `s=2×5+5×1+1×6+6×4+4×3+3×2=63`. Soluţia problemei este însă în acest caz aranjarea `1-2-4-6-5-3-1` pentru care suma este `82`.

50. Să se genereze toate modalitățile de plată a unei sume S de bani, având la dispoziție oricâte monezi din fiecare valoare. Numărul tipurilor de monezi, valorile acestor monezi precum și suma S se citesc de la tastatură.

Exemplu. Pentru 4 tipuri de monezi de valori 1, 3, 5, 10 și suma de plată S=10 se obțin soluțiile:

```
1 monedă de 10
2 monezi de 5
1 monedă de 1, 3 monezi de 3
2 monezi de 1, 1 monedă de 3, 1 monedă de 5
4 monezi de 1, 2 monezi de 3
5 monezi de 1, 1 monedă de 5
7 monezi de 1, 1 monedă de 3
10 monezi de 1.
```

51. La un magazin vânzătorul trebuie să dea unui client un rest de S lei. Cunoscând valorile fiecărui tip de monedă și faptul că vânzătorul nu vrea să dea rest două monezi de aceeași valoare, afișați toate variantele în care vânzătorul poate da restul.

Exemplu. Dacă n=5, valorile monezilor sunt (2,3,5,6,10) iar suma de plată este S=10 atunci singurele soluții sunt:

```
1 monedă de 10 sau
1 monedă de 2 + 1 monedă de 3   1 monedă de 5.
```

52. Modificați programul anterior astfel încât să obțineți numărul maxim de fise pe care le poate da ca rest vânzătorul, în ipoteza că nu va da rest două monezi de același tip.

53. Modificați programul anterior astfel încât să obțineți toate modalitățile în care vânzătorul poate da restul, în ipoteza că poate da rest mai multe fise de același tip iar numărul de monezi din fiecare tip este limitat, numărul de fise din fiecare tip fiind dat de la tastatură.

Exemplu. Dacă există 5 tipuri de monezi de valori 5 (2 bucăți), 10 (3 bucăți), 20 (4 bucăți), 50 (2 bucăți), 100 (3 bucăți), iar suma este S=100 atunci soluțiile sunt:

```
1 monedă de 100
2 monezi de 50
1 monedă de 10, 2 monezi de 20, 1 monedă de 50
2 monezi de 10, 4 monezi de 20
3 monezi de 10, 1 monedă de 20, 1 monedă de 50
2 monezi de 5, 2 monezi de 20, 1 monedă de 50
2 monezi de 5, 1 monedă de 10, 4 monezi de 20
2 monezi de 5, 2 monezi de 10, 1 monedă de 20,
                                    1 monedă de 50
2 monezi de 5, 3 monezi de 10, 3 monezi de 20.
```

54. Modificați programul anterior astfel încât vânzătorul să dea ca rest un număr minim de fise, în ipoteza că poate da rest mai multe fise de același tip iar numărul de monezi din fiecare tip este nelimitat.

Exemplu. Pentru 4 tipuri de monezi de valori 60, 59, 6, 1 și suma de plată S=124 se obține soluția:

```
2 monezi     59, 1 monedă de 6.
```

55. Să se genereze toate șirurile strict crescătoare formate din numere naturale cu proprietatea că primul element din șir este egal cu n, iar ultimul element al șirului este egal n+k. Numerele n și k sunt citite de la tastatură.

Exemplu. Pentru n=7 și k=3 se vor afișa șirurile: (7,8,9,10), (7,8,10), (7,9,10), (7,10).

56. Să se genereze toate șirurile strict crescătoare formate din numere naturale cu proprietatea că primul element din șir este cel puțin egal cu n, iar ultimul element al șirului este cel mult egal n+k. Numerele n și k sunt citite de la tastatură.

Exemplu. Pentru n=7 și k=2 se vor afișa șirurile (7), (7,8), (7,8,9), (7,9), (8), (8,9), (9).

57. N elevi trebuie așezați pe două rânduri. Rândurile nu trebuie să conțină același număr de elevi. Se cere să se afișeze o astfel de așezare încât diferența înălțimii totale a elevilor de pe cele două rânduri să fie minimă.

Exemplu. Dacă înălțimile elevilor sunt: 110, 130, 150, 130 atunci elevii se vor așeza astfel:

Rândul 1: 110, 150

Rândul 2: 130, 130

58. Să se afișeze toate lanțurile ce se pot construi cu cele n piese de domino (citite de la tastatură) știind că fiecare piesă are înscrise în ordine două numere din mulțimea {1,2,3,4,5,6} iar două piese se pot plasa pe lanț în poziții consecutive dacă și numai dacă primul număr înscris pe cea de a doua piesă coincide cu cel de al doilea număr înscris pe prima piesă

Exemplu. n=5 și piesele (1,2) (3,4) (4,2) (1,3) (4,4) se obțin următoarele soluții:

```
1, 3, 5, 2, 4        5, 3, 1, 4, 2
2, 4, 1, 3, 5        1, 4, 2, 5, 3
3, 5, 2, 4, 1        2, 5, 3, 1, 4
4, 1, 3, 5, 2        3, 1, 4, 2, 5
5, 2, 4, 1, 3        4, 2, 5, 3, 1
```

59. Se consideră o bară a cărei lungime se exprimă printr-un număr natural. Să se scrie un program care să determine toate modurile de tăiere a barei în segmente de lungimi date, exprimate prin numere naturale nenule, astfel încât din fiecare segment să existe cel puțin o bucată. Lungimea inițială a barei și lungimile segmentelor în care se taie bara se vor introduce de la tastatură.

Exemplu. Pentru `L=23` și lungimile segmentelor `(1,2,3,5,8)` atunci o soluție este `1x1+3x2+1x3+1x5+1x8=23`.

60. Să se determine toate elementele mulțimii $\{(x,y,z) \in N^3 \mid 3x+y+4z=100\}$.

Exemplu. Trei dintre soluții sunt

`x=6, y=26, z=14`

`x=28, y=8, z=2`

`x=31, y=3, z=1`

61. Pe o mare există `n` porturi (`n<=10`). Staționarea într-un anumit port se taxează cu un anumit cost. Să se stabilească toate voiajele distincte prin `p` porturi (`p<=n`) astfel încât să nu se depășească un cost total dat. Porturile au denumirile de maxim `16` caractere, iar costurile sunt numere reale. Fișierul de intrare este "`port.in`" și are următoarea formă

```
n  p
nume_port_1 cost1

nume_port_n cost_n
cost_total_maxim
```

Afișarea rezultatelor se va face pe ecran.

Exemplu. Dacă fișierul de intrare este

```
8 4
A 10
B 3.5
D 2
E 4.5
F 2.5
H 1.5
I 5
J 3
11
```

atunci soluțiile sunt `(D,F,H,J)`, `(D,F,H,I)`, `(D,E,H,J)`, `(D,E,F,H)`, `(B,F,H,J)`, `(B,D,H,J)`, `(B,D,F,J)`, `(B,D,F,H)`.

62. Un avion al NATO, care transporta ajutoare pe care urma să le paraşuteze refugiaţilor de la graniţele Afganistanului, a suferit o defecţiune neaşteptată, în timpul zborului. Pentru a nu se prăbuşi trebuie să se renunţe la o parte din încărcătura de la bord. Se ştie că greutatea maximă pe care o poate transporta în continuare este **G**. Se cunoaşte greutatea fiecărui pachet în parte, numărul total al pachetelor fiind **n**. De asemenea, numărul minim de pachete (pentru a putea contribui la supravieţuirea populaţiei unei anumite zone ţintă) pe care trebuie să le păstreze este **a**, iar numărul maxim de pachete păstrate este **b** (din lipsă de spaţiu, o parte din cala avionului nemaiputând fi utilizată din cauza defecţiunii apărute). Care sunt posibilităţile existente?

Exemplu. Pentru `n=10`, `G=9`, `a=3`, `b=4` şi greutăţile pachetelor `(1,2,3,4,5,6,7,8,9,10)` se obţin soluţiile `(2,3,4)`, `(1,3,5)`, `(1,3,4)`, `(1,2,6)`, `(1,2,5)`, `(1,2,4)`, `(1,2,3)` (numerele indicând numărul de ordine al pachetului).

63. Un elev împătimit al jocurilor de calculator doreşte să cumpere `3` CD-uri cu jocuri. Pentru aceasta el solicită unei firmă oferta cu toate jocurile pe care le pot inscripţiona pe CD. Oferta conţine, pentru fiecare joc, numele jocului, dimensiunea în Kb ocupată, şi numărul CD-ului pe care este înregistrat jocul în cadrul colecţiei firmei respective. Elevul doreşte să înscrie cât mai multe jocuri pe cele trei CD-uri. Dacă există mai multe variante care satisfac aceste condiţii el va alege acea variantă în care suma spaţiilor neocupate pe cele trei CD-uri este minimă. Se ştie că un CD are o capacitate de `700` Mb (`1Mb=1024` Kb).

Exemplu. Dacă denumirile şi dimensiunile în Kb a jocurilor sunt: `(A,B,C,D,E,F,G,H,I,J,K)` respectiv `(716800,307200,307200, 256000,204800,204800,409600,307200,614400,153600,102400)` atunci soluţia problemei va fi:

```
CD-ul 1: C, D, J          CD-ul 2: E, F, H

CD-ul 3: I, K.
```

64. Se dau coordonatele `(x,y)` a `n` puncte în plan. Să se determine mulţimile de `p` drepte obţinute prin unirea a două din punctele date astfel încât toate punctele de intersecţie a acestor drepte să aparţină mulţimii de puncte dată iniţial.

Exemplu. Dacă punctele considerate au următoarele coordonate `(1,3.5)`, `(7,3.5)`, `(4,5.5)`, `(1,7.5)`, `(7,7.5)` şi `p=2` atunci o soluţie este: segmentul 1: `(7,3.5) – (1,7.5)`, segmentul 2: `(4,5.5) – (7,7.5)`.

65. Să se afişeze toate modurile în care se poate perfora un bilet de autobuz. Un bilet are 9 puncte de perforare posibile:

```
* * *
* * *
* * *
```

66. Se dau coordonatele (x,y) a n puncte în plan (n<=10). Se cere să se atașeze fiecărui punct una din cele m culori disponibile (m<=5), astfel încât oricare două puncte care se văd să fie colorate diferit. Două puncte în plan se văd dacă pe segmentul de dreaptă generat de cele două puncte nu se mai găsește nici un alt punct.

Exemplu. Dacă punctele au următoarele coordonate (1,8), (7,8), (7,5.5), (7,3), (1,3), (4,5.5) atunci ele pot fi colorate cu m=4 culori astfel: (1,2,3,1,2,4), (1,2,4,1,2,3), (2,1,4,2,1,3) etc. (numerele indică culorile punctului respectiv, considerate în ordinea în care au fost date).

67. Se citesc de la tastatură un număr natural n<=15, n numere naturale $x_1, x_2, \ldots, x_n$ mai mici sau egale cu 10000 și un număr natural S cu cel mult 9 cifre. Se cere să se determine șirurile de semne + și - ce trebuiesc puse în fața numerelor $x_1, x_2, \ldots, x_n$ pentru ca rezultatul expresiei obținute să fie egal cu S.

Exemplu. Pentru n=5, x=(4,1,3,8,2), S=10 soluțiile sunt:

```
+--++
-++++
```

68. Un pomicultor vrea să planteze 2n puieți de pomi fructiferi de înălțimi diferite pe două rânduri egale, astfel încât pomii de pe aceeași linie să fie în ordine crescătoare (de la stânga la dreapta) a înălțimilor și în plus fiecare pom din primul rând să fie mai înalt decât pomul corespunzător din linia a doua. Puteți să-i dați o mână de ajutor pomicultorului?

Exemplu. Dacă n=3 și înălțimile pomilor sunt în ordine (5,9,1,2,3,7) soluțiile sunt:

```
Soluția 1:   R1: 2 1 4        Soluția 2: R1: 2 1 5
             R2: 6 5 3                   R2: 6 4 3
Soluția 3:   R1: 2 6 1        Soluția 4: R1: 2 6 4
             R2: 5 4 3                   R2: 1 5 3
Soluția 5:   R1: 2 6 5
             R2: 1 4 3.
```

69. Să se descompună un număr natural n în toate modurile posibile, ca sumă de p numere naturale nenule, distincte. Soluțiile care diferă doar prin ordinea termenilor nu se vor considera distincte.

Exemplu. Pentru n=10 și p=3 soluțiile sunt: (1,2,7), (1,3,6), (1,4,5), (2,3,5).

70. Fiind dat un număr N, să se scrie ca sumă de numere întregi distincte al căror cel mai mare divizor comun este maxim posibil.

Exemplu. Pentru n=68 soluția este (17,51).

71. Să se descompună un număr natural n în toate modurile posibile, ca sumă de p numere naturale nenule. Numerele se pot repeta în cadrul descompunerii. Soluțiile care diferă doar prin ordinea termenilor nu se vor considera distincte.

Exemplu. Pentru n=10 și p=3 soluțiile sunt: (1,1,8), (1,2,7), (1,3,6), (1,4,5), (2,2,6), (2,3,5), (2,4,4), (3,3,4).

72. Să se descompună un număr natural n ca sumă de 3 și 5. Se vor afișa toate soluțiile. Soluțiile care diferă doar prin ordinea termenilor nu se vor considera distincte.

Exemplu. Pentru n=30 soluțiile sunt: (3,3,3,3,3,3,3,3,3,3), (3,3,3,3,3,5,5,5), (5,5,5,5,5,5).

73. Să se descompună un număr natural n ca sumă de numere prime distincte. Se vor afișa toate soluțiile. Soluțiile care diferă doar prin ordinea termenilor nu se vor considera distincte.

Exemplu. Pentru n=20 soluțiile sunt: (7,13), (3,17), (2,7,11), (2,5,13).

74. Să se descompună un număr natural n ca sumă de numere prime. Un număr prim se poate repeta în cadrul descompunerii. Se vor afișa toate soluțiile. Soluțiile care diferă doar prin ordinea termenilor nu se vor considera distincte.

Exemplu. Pentru n=10 soluțiile sunt (2,2,2,2,2), (2,2,3,3), (2,3,5), (3,7), (5,5).

75. Spunem că un număr are proprietatea sufixului, dacă el apare ca sufix al pătratului său, adică cifrele sale apar la sfârșitul șirului de cifre ce reprezintă acel număr ridicat la pătrat. Să se genereze toate numerele având maxim 4 cifre și care au proprietatea sufixului.

Exemplu. 9376 are proprietatea sufixului (9376^2=87909376).

76. Să se precizeze toate modalitățile de repartizare a orelor de matematică (3 ore/săptămână), fizică (2 ore/săptămână) și informatică (3 ore/săptămână) în orarul unei clase știind că în fiecare zi trebuie programată cel puțin una și cel mult trei din aceste ore.

Exemplu. O soluție poate fi:

```
Luni: matematica (2h)
Marți: fizica (1h)
Miercuri: informatica (2h), matematica (1h)
Joi: fizica (1h)
Vineri: informatica (1h)
```

77. Să se scrie numărul N ca sumă de numere întregi pozitive al căror produs este maxim posibil.

Exemplu. Pentru n=10 soluția este (2,2,3,3).

78. Să se afișeze toate modalitățile de scriere a unui număr N ca sumă de numere întregi pozitive, ale căror inverse însumate nu depășesc 1.

Exemplu. Pentru N=5 există o singură astfel de modalitate de scriere N=2+3, $\frac{1}{2}+\frac{1}{3}=\frac{5}{6}<1$. Pentru n=10 soluțiile sunt: (2,8), (3,3,4), (3,7), (4,6), (5,5), (10).

79. Să se programeze m examene în n zile. Se va avea în vedere atât cazul m<=n cât și cazul n<=m.

Exemplu. Dacă n=3 și m=5 o soluție este:

Ziua 1: examenele 1 și 3;

Ziua 2: examenul 2; Ziua 3: examenele 4 și 5.

80. Un examen de admitere constă din n probe. Știind numărul maxim de puncte ce se pot obține la fiecare probă în parte precum și numărul minim de puncte necesar pentru a reuși la examen, să se afișeze toate variantele de reușită.

Exemplu. Dacă n=3, numărul minim necesar de puncte este 35 iar numărul maxim de puncte ce se pot obține la fiecare probă este: 35, 20, 25 atunci o soluție este:

Proba 1: 5 puncte; Proba 2: 10 puncte; Proba 3: 25 puncte

81. Să se determine toate permutările mulțimii {1,2,...,n} astfel încât elementele p, p+1, p+2,...,p+m să rămână în această ordine (chiar dacă nu pe poziții consecutive).

Exemplu. Pentru n=4, p=1, m=1 se obțin soluțiile: (1,2,3,4), (1,2,4,3), (1,3,2,4), (1,3,4,2), (1,4,2,3), (1,4,3,2), (3,1,2,4), (3,1,4,2), (3,4,1,2), (4,1,2,3), (4,1,3,2), (4,3,1,2).

82. Un student la filologie posedă n dicționare bilingve care permit traducerea dintr-o limbă i într-o limbă j (1<=i,j<=m). Să se determine toate seturile de dicționare care-i permit o traducere din limba S în limba D. Se vor afișa toate soluțiile.

Exemplu. Dacă n=8, iar dicționarele sunt (limbile se codifică prin numere dar aici pentru a fi mai sugestiv exemplul am dat chiar denumirile limbilor):

1. Engleză -> Franceză
2. Română -> Spaniolă
3. Franceza -> Germană
4. Germană -> Italiană
5. Spaniolă -> Engleză
6. Germană -> Engleză
7. Română -> Germană
8. Italiană -> Engleză

atunci pentru a traduce din limba Română în limba Franceză se utilizează dicționarele 7, 4, 8, 1.

83. La un concurs "LOTO" s-au extras 6 numere între 1 şi 49 (binecunoscutul joc 6 din 49). Se ştie că ordinea numerelor în cadrul extragerii este importantă. Fiecare participant la joc a avut înscrise de asemenea câte 6 numere între 1 şi 49 pe biletele lor, şi de această dată ordinea numerelor fiind importantă. Doi participanţi la joc au reuşit să ghicească (pe poziţia corectă) n1 respectiv n2 numere, şi au mai ghicit m1 respectiv m2 numere însă pe alte poziţii decât cele în care au fost acestea extrase. Afişaţi din toate extragerile posibile (împreună cu numerele de pe taloanele celor doi jucători) pe acelea care îndeplinesc condiţiile problemei.

Exemplu. Dacă n1=2, n2=1, m1=1, m2=1, atunci o soluţie poate fi:

```
Numere extrase:        12,  1,  6,  8, 35,  4
Primul jucător:        46,  1, 40, 12, 18,  4
Al doilea jucător:     35,  5,  6,  9, 32, 16
```

84. Se dă o mulţime $X=\{x_1,x_2,x_3,\ldots,x_n\}$ de numere întregi şi n submulţimi $A_1,A_2,\ldots,A_m$, cu proprietatea $A_1\cup A_2\cup\ldots\cup A_m=X$ şi $A_i\not\subset A_j$ pentru $i\neq j$. Să se aleagă cât mai puţine submulţimi din cele m date astfel încât reuniunea lor să fie tot X (acoperirea minimală a mulţimii X).

Exemplu. Pentru mulţimea $X=\{1,2,3,4,5\}$ şi submulţimile $A_1=\{1,2,3\}$, $A_2=\{1,2,4\}$, $A_3=\{1,3,4\}$, $A_4=\{1,3,5\}$ şi $A_5=\{1,4,5\}$ o acoperire minimală a lui X este (A_2,A_4).

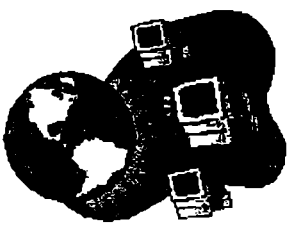
85. Să se scrie un program care, citind un cuvânt şi un număr natural cuprins între 1 şi lungimea cuvântului, să afişeze toate anagramările obţinute din cuvânt, după eliminarea literei de pe poziţia citită.

Exemplu. Dacă cuvântul este 'lemn' iar numărul citi este 3 se obţin următoarele soluţii: len, lne, eln, enl, nle, nel.

86. Se citesc de la tastatură n cuvinte având toate aceeaşi lungime de m caractere, şi două cuvinte c1 şi c2 ambele de lungime p. Să se afişeze toate modalităţile de alegere a k cuvinte din cele n date astfel încât scriind cuvintele unul sub altul să se poată forma pe prima coloană cuvântul c1 iar pe ultima coloană cuvântul c2. Dacă există mai multe soluţii se vor afişa toate.

Exemplu. Pentru n=10, m=3, c1='pitic', c2='micut' şi cuvintele 'toc', 'are', 'pom', 'imi', 'par', 'cat', 'ele', 'tic', 'itu', 'cot' o soluţie este (nu singura):

```
p o m
i m i
t o c
i t u
c o t
```



87. Se dau n cuvinte, formate din litere mici ale alfabetului englez. Să se formeze cuvintele de lungime maximă, folosind litere din aceste cuvinte, respectând următoarele reguli:

a i-a literă a cuvântului este o literă de pe poziția i a unui cuvânt dat;

literele fiecărui cuvânt sunt ordonate lexicografic.

Exemplu. Pentru cuvintele `'la'`, `'revedere'`, `'turda'` se obțin cuvintele `'luv'`, `'ruv'`, `'tuv'`

88. La startul unui concurs sunt prezenți n concurenți. Să se afișeze toate posibilitățile de a intra în concurs a concurenților (concurenți intră în concurs pe rând) în ipoteza că din motive mai mult sau mai puțin obiective concurentul cu codul c_1 trebuie să intre în concurs între primii 3 iar concurentul c_2 trebuie să intre în concurs între ultimii 2.

Exemplu. Dacă n=4, c1=2, c2=1 atunci soluțiile sunt: (2,3,1,4), (2,3,4,1), (2,4,1,3), (2,4,3,1), (3,2,1,4), (3,2,4,1), (3,4,2,1), (4,2,1,3), (4,2,3,1), (4,3,2,1).

89. Un halterofil de performanță dorește să urmeze un program de pregătire având la dispoziție n greutăți de mase distincte două câte două. Antrenamentul din fiecare zi este împărțit în m părți, în fiecare parte sportivul putând ridica o singură greutate. Pentru ca antrenamentul să fie eficient, halterofilul își propune să ridice greutățile astfel:

i) În prima parte a antrenamentului el poate ridica orice greutate (numai una);

ii) Pentru oricare două părți i și j, consecutive ale antrenamentului, greutatea ridicată în j trebuie să fie mai mare, sau cel puțin egală cu greutatea ridicată în partea i;

iii) Pentru oricare trei părți consecutive ale antrenamentului greutatea ridicată în cea de a treia parte trebuie să fie strict mai mare decât cea ridicată în prima parte.

Se cere să se afișeze greutățile pe care le ridică sportivul în fiecare zi.

Exemplu. Dacă n=5, m=3 iar greutățile având masele de 10, 15, 50, 25, 20 soluțiile sunt:

```
10 10 15
10 10 20
10 10 25
10 10 50
10 15 15
10 15 20
10 15 25
10 15 50
10 20 20
10 20 25
10 20 50
etc
```

90. Proprietarul unui magazin trebuie să se aprovizioneze de la un magazin comercial en-gros. Produsele existente în depozit sunt numerotate cu 1,2,..,n. Pentru fiecare produs se cunoaşte categoria (grupa) din care face parte, codificată printr-un caracter (de exemplu 'a'=produse alimentare, 'c'=cosmetice, etc). Determinaţi toate variantele pe care le are proprietarul de a alege produsele pe care la va cumpăra, ştiind că, dintr-o categorie, indiferent care va fi aceasta, va cumpăra cel mult b produse (unde b este cunoscut).

Exemplu. Dacă n=10 şi codurile produselor sunt a, c, a, a, a, b, c, b, p, c, iar b=2 câteva din soluţiile problemei sunt:

1, 2, 3, 10

2, 3, 4, 8, 9 etc.

91. La curtea regelui Arthur sunt n cavaleri, fiecare cavaler având duşmani. MERLIN trebuie să-i aşeze la MASA ROTUNDĂ, astfel încât doi duşmani să nu fie vecini. Să se determine toate modalităţile de aşezare a cavalerilor la masă.

Exemplu. Dacă n=6 iar duşmani sunt (1 cu 3), (2 cu 4), (1 cu 5) şi (4 cu 6) atunci o modalitate de aşezare a cavalerilor este 3, 4, 5, 2, 1, 6, 3.

92. La un banchet participă mai mulţi elevi ai Colegiului Naţional "Gheorghe Lazăr" Sibiu. Se ştie că elevii fac parte din k clase paralele. Pentru fiecare clasă se cunoaşte numărul de fete şi numărul de băiaţi. De asemenea se cunoaşte că la Restaurantul "Împăratul Romanilor" mesele au fost aranjate sub forma unei mese rotunde la care au loc toţi elevii participanţi. Să se afişeze (dacă este posibil) toate modalităţile de a aranja elevii la masă astfel încât:

fiecare fată să aibă drept vecini de masă doi băieţi;

nici un elev nu va avea vecini la masă colegi de-ai săi de clasă.

Exemplu. Dacă avem 3 clase cu următoarea componenţă:

```
Prima clasă:        Fete: A, B, C
                    Băieţi: D, E, F, G, H, I
A doua clasă:       Fete: J, K
                    Băieţi: L, M, N
A treia clasă:      Fete: O, P, R
                    Băieţi: S, T, U, V
```

atunci o posibilă aşezare la masă este: A – S – M – B – T – N – C – U – D – V – J – E – K – F – O – H – P – G – R – I – L – A.

93. Într-un grup de persoane, fiecare persoană se cunoaşte pe sine şi cunoaşte eventual şi alte persoane din grup. Să se formeze şi să se afişeze toate echipele posibile de persoane astfel încât, pentru o echipă, fiecare persoană să fie cunoscută de cel puţin un membru al echipei.

94. Un dresor trebuie să scoată `m` lei şi `n` tigri din arenă, astfel încât să nu scoată doi tigri unul după altul. Să se genereze toate posibilităţile de înşiruire a leilor şi tigrilor.

Exemplu. Dacă `m=5 lei` şi `n=3 tigri`, o soluţie este: `leu – tigru – leu – leu – leu – tigru – leu – tigru`.

95. O caravană formată din `n` cămile călătoreşte prin deşert, în şir indian. Pentru a sparge monotonia zilelor lungi de drum, beduinul şef se hotărăşte să schimbe aşezarea cămilelor, astfel încât fiecare cămilă să nu mai vadă în faţa ei aceeaşi cămilă de până atunci. Să se genereze toate posibilităţile de aşezare a cămilelor, cunoscând modul de aşezare din prima zi.

Exemplu. Dacă `n=5` iar aşezarea din prima zi este `1,2,3,4,5` atunci câteva soluţii sunt:

```
1, 3, 2, 5, 4
2, 1, 4, 3, 5
5, 4, 3, 2, 1
```

etc.

96. Fie `n` persoane, `n` locuri de muncă şi `C(i,j)` costul angajării persoanei `i` la locul de muncă `j`. Să se determine modul optim de angajare a tuturor persoanelor.

97. Scrieţi un program care să aşeze pe o tablă de şah `n×n`, `n` cai, fiecare cal pe o altă coloană, fără să existe atac între ei.

Exemplu. Pentru n=5 o soluţie poate fi:

```
• – – – •
– • – – –
– – – – –
– – – • –
– – • – –
```

unde cu • am marcat poziţiile cailor.

98. Pe o tablă de şah cu `n` coloane şi `m` linii se cere să se aşeze `n` piese, fiecare pe câte o coloană, care să nu se atace între ele. Piesele se atacă precum o tură, un nebun şi un cal simultan.

99. Determinaţi toate funcţiile surjective `f:{1,2,…,n}→{-1,0,1}` astfel încât:

$$f^2(1) + f^2(2) + \quad + f^2(n) = m,$$

`m` fiind dat.

Exemplu. Pentru `n=5, m=3` trei dintre soluţii sunt (mai există şi altele):

```
f(1)= -1, f(2)=0, f(3)=1, f(4)= -1
f(1)= -1, f(2)=0, f(3)=1, f(4)= 1
f(1)= 1, f(2)=0, f(3)= -1, f(4)= -1
```

100. Se dă un vector $(x_1, x_2, \ldots, x_n)$ de numere reale. Să se determine toți subvectorii crescători de lungime maximă ai acestui vector:

$$x_{i1}, x_{i2}, \ldots, x_{ik} \qquad i1<i2<\ldots<ik \quad x_{i1}<=x_{i2}<=\ldots<=x_{ik}.$$

Exemplu. Pentru n=8 și vectorul x=(5,1,7,2,9,3,14,18) se obțin următoarele soluții:

```
5, 7, 9, 14, 18
1, 7, 9, 14, 18
1, 2, 9, 14, 18
1, 2, 3, 14, 18
```

101. Se dă numărul natural n>0. Să se determine toate șirurile de n paranteze care se închid corect.

Exemplu. Pentru n=6 avem soluțiile ((())), ()()(), (()()), ()(()), (())().

102. Fie n segmente situate pe o aceeași dreaptă, numerotate distinct, de la 1 la n. Să se scrie un program care determină numărul minim de culori necesare pentru a colora cele n segmente astfel încât oricare două segmente care se intersectează să fie colorate diferit și, de asemenea, să se determine o astfel de colorare. Pentru fiecare segment de dreaptă se dă extremitatea inițială a segmentului și lungimea sa.

Exemplu. Pentru n=5, extremitățile inițiale (9,2,3,1,11) și lungimile (3,4,6,3,4) se obține soluția

```
Segmentul 1 culoarea 1
Segmentul 2 culoarea 2
Segmentul 3 culoarea 3
Segmentul 4 culoarea 1
Segmentul 5 culoarea 2
```

Există 36 de soluții.

103. Fie P_1, P_2, , P_n o mulțime de programe având lungimile L_1, L_2, , L_n, ce trebuie inserate pe o bandă de lungime L, astfel încât $L_1+L_2+\ldots+L_n \geq L$. Să se determine o submulțime a mulțimii programelor, maximală din punct de vedere al numărului de programe conținute, care să fie memorate pe bandă, în același timp realizându-se minimizarea spațiului rămas pe bandă.

Exemplu. Dacă n=6, L=10, și lungimile programelor sunt (5,3,6,2,9,4) soluția este

```
Programul 1 de lungime 5
Programul 2 de lungime 3
Programul 4 de lungime 2
Spațiul liber rămas este 0
```

104. Un elev din clasa I-a are la dispoziție `n` litere mici din alfabetul limbii engleze. Doamna învățătoare îi cere următoarele lucruri:

a) Să verifice dacă există litere care apar de mai multe ori și să rețină toate literele distincte o singură dată;

b) Să aşeze aceste litere în ordine alfabetică (fie `m` numărul lor);

c) Să formeze toate cuvintele posibile din `k` litere (`k≤m`) în care literele sunt ordonate alfabetic în cadrul cuvântului, iar o literă apare în cuvânt o singură dată;

d) Să formeze toate cuvintele din `k>=2` litere în care literele sunt ordonate alfabetic în cadrul cuvântului, iar o literă se poate repeta de cel mult `p` ori.

Exemplu. Pentru `n=7`, `k=4`, `p=2`, și literele `a, h, m, a, e, a, i` se va afișa:

a) `a h m e i`

b) `a e h i m`

c) `aehi, aehm, aeim, ahim, ehim`

d) `aaee, aaeh, aaei, aaem, aahh, aahi, aahm, aaii, aaim, aamm, aeeh, aeei, aeem, aehh`

105. Pentru un număr natural `n` dat de la tastatură, să se afișeze toate șirurile formate din exact `n` litere, fiecare șir generat având următoarele proprietăți:

este format doar din litere mari ale alfabetului latin;

toate literele din șir sunt distincte;

oricare două litere alăturate din șir sunt alăturate și în alfabet.

Exemplu. Pentru `n=5` se obțin: `'ABCDE'`, `'BCDEF'`, `'GHIJK'`,

106. Dintr-un număr de cuvinte citite de la intrare, urmează să se formeze fraza cea mai lungă cu putință, respectând următoarele reguli:

cuvintele vor fi aşezate în frază astfel încât litera cu care se termină un cuvânt să fie litera cu care începe următorul cuvânt;

în frază cuvintele vor fi despărțite de un blanc.

Se cere să se listeze fraza de lungime maximă. Prin frază de lungime maximă vom înțelege:

a) o frază cu număr maxim de caractere (nu se numără și spațiile);

b) o frază cu număr maxim de cuvinte.

Exemplu. Dacă cuvintele sunt 'elev', 'viitor', 'vin', 'noapte', 'etaj', 'palat', 'roua', 'apartament', 'joc', 'arici', 'ic' atunci frazele afișate vor fi:

a) `vin noapte elev viitor roua apartament`

b) `vin noapte elev viitor roua arici ic`

107. Pentru alcătuirea unei garnituri de tren, regulamentul prevede următoarele:

fiecare vagon restaurant trebuie să fie încadrat la stânga şi la dreapta de câte cel puţin un vagon de călători;

fiecare vagon de marfă este urmat (în sensul de la locomotivă spre ultimul vagon) de cel puţin un vagon de călători.

Fie `n` (`n<=10`) numărul de vagoane ce alcătuiesc garnitura de tren (`n` este citit de la intrare). Se cer următoarele:

1) să se scrie într-un fişier cu numele `f.dat` variantele distincte de garnituri ce pot fi alcătuite şi în continuare numărul acestor variante;

2) Modificaţi programul prin introducerea unei noi restricţii:

numărul de vagoane restaurant este limitat superior de `[n/4]`.

orice garnitură trebuie să aibă cel puţin un vagon marfă în componenţă.

Exemplu. Pentru `n=4` se obţin următoarele soluţii:

1) `21` soluţii

```
restaurant, persoane, restaurant, persoane
restaurant, persoane, marfa, persoane

persoane, persoane, persoane, persoane
```

2) `13` soluţii

```
restaurant, persoane, marfă, persoane
restaurant, persoane, persoane, marfă

persoane, persoane, persoane, marfă
```

108. Se consideră `n` cuvinte. Se cere să se determine cel mai lung cuvânt format prin alipirea cuvintelor date care respectă condiţia: nici o literă nu va apărea în cadrul cuvântului obţinut de mai mult de `p` ori.

Exemplu. Pentru `n=6`, `p=3` şi cuvintele `'acasa'`, `'masa'`, `'elicopter'`, `'cartof'`, `'pantof'`, `'prostioara'` soluţia se obţine din alăturarea cuvintelor `'elicopter-cartof-pantof'`

109. Doi fraţi, Andrei şi Bogdan doresc să-şi împartă mai multe cadouri. Fiecare cadou va fi dat fie lui Andrei fie lui Bogdan şi nici unul din cadouri nu poate fi împărţit. Fiecare cadou are o valoare întreagă pozitivă. Fie `A` şi `B` valorile totale ale cadourilor primite de Andrei, respectiv Bogdan. Scopul este de a minimiza valoarea absolută a diferenţei `A-B`. Scrieţi un program care calculează valorile `A` şi `B`.

Exemplu. Pentru `5` cadouri având valorile `12, 5, 23, 11, 9` soluţia este de a da lui Andrei cadourile cu valorile `12, 5` şi `11` iar lui Bogdan cadourile cu valorile `23` şi `9`.

110. Pentru elaborarea unui test de aptitudini se dispune de un set de n întrebări, întrebarea i fiind cotată cu p_i puncte. Se cere să se elaboreze toate chestionarele având între a şi b întrebări şi totalizând între c şi d puncte.

Exemplu. Dacă n=8, punctajele întrebărilor (10, 15, 10, 25, 5, 15, 5, 10), a=2, b=4, c=20, d=30, atunci câteva soluţii sunt:

1, 2
1, 3, 5
1, 2, 5
1, 3, 5, 7 etc.

111. Se dau n camere cu proprietatea că din camera i se poate trece doar în camera i+1, $1 \le i \le n-1$, în fiecare găsindu-se m_i mere şi p_i pere, cu excepţia ultimei camere n, pentru care $m_n=p_n=0$.

O persoană cu un rucsac gol şi suficient de încăpător trece pe rând prin fiecare cameră. În prima cameră încarcă în rucsac toate merele sau toate perele şi trece în camera a doua. Începând cu aceasta, în fiecare cameră, cu excepţia ultimei, persoana descarcă rucsacul, se odihneşte, încarcă fie toate merele fie toate perele şi trece în camera următoare. Să se afişeze toate posibilităţile de alegere a fructelor din fiecare cameră, cantitatea de fructe care ajunge în ultima cameră şi felul acestora, precum şi numărul total de calorii consumate, ştiind că pentru transportul unui fruct între două camere se consumă o calorie.

Exemplu. Pentru n=6 şi conţinutul din fiecare cameră următorul: (5 mere,3 pere), (3 mere,2 pere), (10 mere,25 pere), (5 mere,5 pere), (17 mere,12 pere), atunci o parte din soluţii sunt:

(mere,mere,mere,mere,mere) în final => 30 de mere, consum de calorii 84

(mere,mere,mere,mere,pere) în final => 12 pere, consum de calorii 66

(pere,pere,mere,pere,mere) în final => 7 mere, consum de calorii 30 ete.

112. Să se modifice programul anterior astfel încât să se afişeze acea soluţie pentru care consumul de calorii este minim.

Exemplu. Pentru aceleaşi date de intrare se obţine soluţia:

(mere,pere,mere,pere,mere) în final => 7 mere, consum de calorii 29.

113. Să se modifice programul anterior astfel încât în final să avem un număr cât mai mare posibil de fructe. Dacă există mai multe astfel de soluţii se va alege aceea cu număr minim de calorii.

Exemplu. Pentru datele de intrare de mai sus soluţia este

(pere,pere,pere,pere,pere)

consum de calorii 120, număr total de fructe în final 47.

114. Se dă o mulțime A de obiecte cărora li se cunosc valorile și greutățile. Să se găsească o submulțime de valoare maximă care să nu depășească o anumită greutate admisă.

Exemplu. Pentru n=5, greutățile obiectelor (8,3,12,5,5), valorile lor (5,8,2,3,19) iar greutatea maximă admisă de 20, se vor selecta obiectele 1, 2 și 5, greutatea lor totală va fi 16, iar valoarea totală va fi 32.

115. Se consideră o matrice A cu m linii și n coloane. Să se determine toate posibilitățile de a elimina linii din matrice astfel încât matricea rămasă să aibă elementele de coloana k ordonate crescător.

Exemplu. Pentru matricea $A = \begin{pmatrix} 5 & 1 & 7 & 3 \\ 8 & 9 & 2 & 6 \\ 1 & 5 & 4 & 6 \\ 1 & 2 & 3 & 4 \\ 6 & 7 & 9 & 3 \end{pmatrix}$ și k=2 se obțin soluțiile (indicăm liniile care se elimină în cazul fiecărei soluții în parte): (2,3,4,5), (2,3,4), (1,3,4,5), (3,4,5), (2,4,5), (2,4), (2,3,5), (2,3), (1,2,4,5), (1,2,4), (1,2,3,5), (1,2,3), (1,2,3,4).

116. Se dă o scară cu n trepte. Știind că un om poate să urce cel mult 3 trepte odată, să se afișeze toate posibilitățile de a urca scara, și apoi afișați din aceste variante o variantă care necesită un efort minim. Efortul pentru urcarea unei trepte este k1, efortul necesar pentru urcarea a două trepte este k2, iar pentru urcarea a trei trepte odată este k3.

Exemplu. Pentru n=5 trepte și k1=1, k2=3, k3=2 se va afișa (nu neapărat în această ordine)

```
1 1 1 1 1
1 1 1 2
1 1 2 1
1 1 3
1 2 1 1
1 2 2
1 3 1
2 1 1 1
2 1 2
2 2 1
2 3
3 1 1
3 2
```

iar soluția care necesită efort minim este (de exemplu) 1 1 3.

117. Se consideră o matrice A cu m linii şi n coloane. Să se determine numărul minim de linii care trebuiesc eliminate din matrice astfel încât matricea rămasă să aibă elementele de coloana k ordonate crescător, şi să se afişeze o astfel de soluţie.

Exemplu. Pentru cazul de la problema anterioară oricare din soluţiile (2,3) şi respectiv (2,4) este corectă.

118. Se dau două şiruri de numere întregi $A=(a_1,a_2,\dots,a_m)$ şi $B=(b_1,b_2,\dots,b_n)$. Să se genereze, dacă este posibil, toate matriccle cu m linii şi n coloane cu următoarele proprietăţi:

suma elementelor de pe linia i a matricei este egală cu a_i

suma elementelor de pe coloana j a matricei este egală cu b_j

elementele din matrice sunt distincte două câte două şi strict pozitive.

Exemplu. Pentru m=3, a=(31,44,25), n=4, b=(25,35,20,20) se obţin 21008 soluţii printre care:

$$\begin{pmatrix} 1 & 10 & 11 & 9 \\ 19 & 12 & 6 & 7 \\ 5 & 13 & 3 & 4 \end{pmatrix}, \begin{pmatrix} 2 & 6 & 9 & 14 \\ 19 & 12 & 8 & 5 \\ 4 & 17 & 3 & 1 \end{pmatrix}, \begin{pmatrix} 5 & 19 & 6 & 1 \\ 9 & 13 & 10 & 12 \\ 11 & 3 & 4 & 7 \end{pmatrix}.$$

119. La un concurs sportiv s-au înscris n concurenţi, având numerele de concurs 1,2,...,n. Pentru fiecare sportiv se cunoaşte ţara de origine (un şir de caractere). În fiecare zi a competiţiei, vor intra în concurs m concurenţi (m<=n). Afişaţi toate posibilităţile de a stabili ordinea intrării în concurs a celor m concurenţi, ştiind că:

doi sportivi din aceeaşi ţară nu pot evolua unul după altul;

în cadrul unei zile concurenţii intră în concurs în ordinea numărului de concurs.

Exemplu. Pentru n=10, m=3 şi următoarele ţări: România, Franţa, Franţa, Germania, Franţa, România, Grecia, Italia, Italia, Germania, câteva soluţii sunt (în fiecare paranteză apar numerele de concurs ale concurenţilor care vor concura într-o zi):

(1,2,4), (3,6,7), (5,8,10), (9)

(1,2,4), (3,6,7), (5,9,10), (8)

(1,2,8), (4,6,9), (3,7,10), (5)

(1,3,4), (2,7,9), (5,6,10), (8) etc.

120. Se citesc de la tastatură un şir de n numere naturale. Să se scrie un program care afişează lungimea celui mai lung subşir inclus în această şirul dat, cu proprietatea că suma elementelor sale este divizibilă cu 3. Prin subşir se va înţelege o succesiune de elemente din şir nu neapărat adiacente.

Exemplu. Pentru n=5 şi vectorul (1,4,5,6,9) soluţia este (4,5,6,9) sau (1,5,6,9).

121. În figura de mai jos apare un pătrat conținând cifre, astfel încât numerele de 5 cifre corespunzătoare fiecărei linii, fiecărei coloane și fiecăreia dintre cele două diagonale sunt numere prime. Pentru linii și diagonale cifrele se citesc de la stânga la dreapta, iar pentru coloane de sus în jos.

1	1	3	5	1
3	3	2	0	3
3	0	3	2	3
1	4	0	3	3
3	3	3	1	1

Folosind fișierul de intrare `INPUT.TXT`, scrieți un program care să construiască astfel de pătrate de ordinul 5, cu respectarea condițiilor:

suma cifrelor din fiecare dintre cele 12 numere este aceeași (11 pentru exemplul considerat);

cifra din colțul din stânga-sus este dată la intrare (1 în exemplul considerat);

un același număr prim poate apare de mai multe ori în aceeași soluție;

dacă există mai multe soluții, vor fi generate toate;

un număr prim de 5 cifre are prima cifră nenulă (de exemplu 00003 nu este un număr prim de 5 cifre).

Intrarea. Fișierul `INPUT.TXT` conține două linii. Pe prima linie apare suma predefinită a cifrelor fiecăruia dintre numerele prime, iar pe a doua apare cifra din colțul stânga-sus.

Ieșirea. Se face în fișierul text `OUTPUT.TXT`. Fiecare soluție ocupă 5 linii, pe fiecare linie apărând numărul prim corespunzător liniei, format din 5 cifre, scrise fără separator între ele. Soluțiile apar într-o ordine oarecare și sunt despărțite între ele printr-o linie goală.

Exemplu. Pentru situația din figură, fișierul de intrare are forma:

```
11
1
```

122. Se consideră `n` puncte în plan, de coordonate reale, `(X1,Y1)`, `(X2,Y2)`, . . , `(Xn,Yn)`. Elaborați un program care selectează din aceste puncte vârfurile unui pătrat, ce conține numărul maximal de puncte din cele date. Afișați coordonatele punctelor selectate și numărul de puncte incluse în pătratul respectiv.

123. Să se afișeze toate numerele de 9 cifre distincte (fără cifra 0) care înmulțite cu cifra `c` dau un număr de 9 cifre distincte (de asemenea fără cifra 0).

Exemplu. Pentru `c=2` o astfel de soluție este numărul `123456789` (`123456789*2=246913578`).

124. Un turist european dorește să viziteze câteva orașe din țările spațiului Schengen. Se cunosc pentru fiecare oraș numărul de obiective turistice existente. Turistul dorește:

124. Un turist european doreşte să viziteze câteva oraşe din ţările spaţiului Schengen. Se cunosc pentru fiecare oraş numărul de obiective turistice existente. Turistul doreşte:

să viziteze cel puţin `k1` şi cel mult `k2` oraşe,

să treacă o singură dată prin fiecare oraş;

numărul de obiective vizitate în prima jumătate a turneului său european să fie mai mare decât numărul obiectivelor vizitate în a doua jumătate;

să viziteze cel puţin `p` obiective turistice.

Ajutaţi turistul nostru în alcătuirea unui itinerar convenabil lui. Se ştie că există drum direct între oricare două oraşe avute în atenţie.

Exemplu. Dacă `n=7`, `k1=2`, `k2=4`, `p=15`, iar numerele de obiective din fiecare oraş sunt `(7,4,10,8,12,3,6)` atunci câteva soluţii sunt:

```
3, 7, 4, 1          7, 5, 2
5, 3, 2, 7          7, 5, 6, 4
7, 5                7, 6, 1
```

125. Acelaşi turist plimbăreţ doreşte anul următor să facă un turneu în America de Sud. Dorinţele lui sunt acum următoarele:

oraşul cu numărul de ordine `x` trebuie obligatoriu vizitat întrucât acolo locuieşte prietenul său din copilărie pe care doreşte neapărat să-l viziteze;

numărul total al obiectivelor turistice vizitate să fie cel puţin `p`, ştiind numărul de obiective turistice din fiecare oraş;

de această dată, din cauza junglei, nu există drum direct între oricare două oraşe. Se dau de la tastatură toate perechile de oraşe între care există drum direct;

să nu treacă de două ori prin acelaşi oraş;

Ajutaţi-l şi de această dată pe turistul nostru.

126. Un grup de prieteni doresc să plece în excursie. După ce au cumpărat o serie de conserve pentru a-şi asigura hrana pe drum, doresc să şi le împartă în aşa fel încât ei să aibă o greutate egală de hrană de cărat. Se cunoaşte: numărul `n` de prieteni, numărul `m` de pachete, greutăţile fiecărui pachet în parte. Să se găsească o posibilitate de împărţire a pachetelor între prieteni astfel încât ei să aibă de cărat EXACT aceeaşi greutate, iar greutatea totală să fie maximă. O parte din pachete pot fi lăsate acasă.

Exemplu. Pentru `n=2` prieteni, `m=5` pachete, şi greutăţile pachetelor `(11,25,20,9,8)` soluţia este: Primul prieten va duce pachetele `1` şi `4`, iar prietenul al doilea pachetul `3`.

127. Între două maluri ale unei văi adânci s-a construit un pod suspendat format din N bucăți de scândură, legate cu liane. Vom considera că scândurile sunt numerotate de la 1 la N, începând de pe malul pe care ne aflăm. În timp unele bucăți de scândură s-au deteriorat, iar altele chiar au dispărut. Pentru traversarea podului se știe că:

- se pot face pași doar de lungime 1, 2 sau 3;
- scândurile deteriorate sunt nesigure, deci pe ele și de pe ele se pot face doar pași de lungime 1.
- evident, nu se poate păși pe o scândură care lipsește.

Scrieți un program care afișează toate modalitățile de traversare a podului (mai exact, de a ajunge pe celălalt mal). La afișarea unei soluții se vor indica numerele de ordine ale scândurilor pe care se calcă.

Exemplu. Pentru N=10, 2 scândurile 2 și 7 lipsă și scândura 5 deteriorată există un număr de total de 48 de posibilități, o soluție fiind (3,6,8). Pentru N=6, scândurile lipsă 2 și 4, și scândura 3 deteriorată, nu există nici o soluție.

128. Se dau $\frac{n(n+1)}{2}$ numere întregi strict pozitive, mai mici sau egale cu 10000. Să se aşeze aceste numere, dacă este posibil, într-o piramidă, cu următoarele proprietăți:

piramida are n nivele, pe fiecare nivel i sunt i numere;

fiecare număr din piramidă, în afară de cele dispuse pe ultimul nivel, este egal cu suma celor două numere pe care se sprijină.

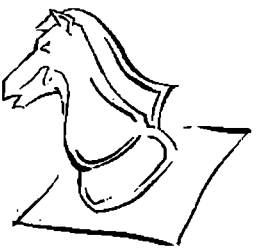
Exemplu. Pentru n=4 și valorile 1,1,2,3,3,4,5,8,9,17 o soluție este:

```
        17
      8   9
    3   5   4
  1   2   3   1
```

129. În secolul XXI, locul clasicelor plăcuțe indicatoare care ajută să ne orientăm în intersecții, este luat de panouri electronice luminoase. Primarul Sibiului a achiziționat trei tipuri de astfel de panouri, de culoare roșie, albastră și verde. El dorește să plaseze câte unul în fiecare intersecție, dar așa încât pe fiecare stradă, delimitată la capete de două intersecții, să nu se întâlnească același tip de panou. Găsiți o soluție de aranjare a panourilor în intersecții conform dorinței primarului. Presupunem că s-a achiziționat un număr suficient de panouri din fiecare culoare. Se cunosc: numărul n al intersecțiilor, numărul m

al străzilor, precum şi m perechi de numere întregi, fiecare pereche reprezentând cele două intersecţii care delimitează strada.

7.2. Backtracking generalizat



130. Pe o tablă de şah de dimensiune `n` se află un cal în poziţia `(x0,y0)`. Afişaţi toate posibilităţile calului de a parcurge toată tabla de şah, fără a trece de două ori prin aceeaşi poziţie.

Exemplu. Pentru `n=5` şi poziţia iniţială `(1,1)` o soluţie este următoarea:

```
 1  18  5  10  3
 6  11  2  19  14
17  22  13  4  9
12  7  24  15  20
23  16  21  8  25
```

131. Pe o tablă de şah de dimensiune `n` se află un cal în poziţia `(x0,y0)`. Cunoscând de câte ori trece calul prin fiecare căsuţă a tablei, afişaţi toate posibilităţile calului de a parcurge această tablă de şah.

Exemplu. Dacă `n=5`, `x0=1`, `y0=1` iar numărul de treceri prin fiecare căsuţă este dat de matricea $\begin{pmatrix} 1 & 0 & 2 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 2 & 0 & 0 \end{pmatrix}$ atunci una dintre soluţii este (există `31` de astfel de soluţii): `(1,1) -> (2,3) -> (4,2) -> (2,1) -> (1,3) ->(2,1)->(1,3)->(3,4)->(5,3)->(3,2)->(5,3)->(4,5)->(3,3)`

132. Pe o tablă de şah de dimensiune `n` se află un cal în poziţia `(x0,y0)`. Cunoscând că prin fiecare căsuţă `(i,j)` calul trece cel puţin o dată şi cel mult de `a[i,j]` ori, afişaţi toate posibilităţile calului de a parcurge tabla de şah.

Exemplu. Dacă `n=4`, `x0=1`, `y0=1` şi prin fiecare căsuţă poate trece de cel mult 2 ori atunci trei dintre soluţii sunt:

```
(1,1),(2,3),(1,1),(2,3),(3,1),(1,2),(2,4),(1,2),(2,4),
(3,2),(1,3),(2,1),(3,3),(1,4),(2,2),(4,3),(2,2),(4,1),
(3,3),(2,1),(4,2),(3,4),(1,3),(3,2),(4,4)
```

(1,1),(2,3),(1,1),(2,3),(3,1),(1,2),(2,4),(1,2),
(2,4),(3,2),(1,3),(2,1),(3,3),(1,4),(3,3),(4,1),(2,2),
(4,3),(2,2),(3,4),(4,2),(2,1),(4,2),(3,4),(1,3),(3,2),
(4,4)

(1,1),(2,3),(1,1),(2,3),(3,1),(1,2),(2,4),(1,2),
(2,4),(3,2),(1,3),(2,1),(3,3),(1,4),(3,3),(4,1),(2,2),
(4,3),(3,1),(4,3),(2,2),(3,4),(4,2),(2,1),(4,2),(3,4),
(1,3),(3,2),(4,4)

133. Pe o tablă de şah de dimensiune `n` se află un cal în poziţia `(x0,y0)`. Afişaţi toate posibilităţile calului de a ajunge în poziţia `(x1,y1)`, fără a trece de două ori prin aceeaşi poziţie.

Exemplu. Pentru `n=5`, `x0=1`, `y0=1`, `x1=5`, `y1=5` câteva din soluţii sunt:

(1,1),(2,3),(1,5),(3,4),(1,3),(2,1),(3,3),(1,2),(2,4),
(3,2),(4,4),(5,2),(3,1),(4,3),(5,5)

(1,1),(2,3),(1,5),(3,4),(1,3),(2,1),(3,3),(1,2),(2,4),
(3,2),(5,1),(4,3),(5,5)

(1,1),(2,3),(1,5),(3,4),(1,3),(2,1),(3,3),(5,4),(3,5),
(1,4),(2,2),(4,1),(5,3),(3,2),(4,4),(5,2),(3,1),(4,3),
(5,5)

(1,1),(3,2),(5,1),(4,3),(5,5)

134. Pe o tablă de şah de dimensiune `n` se află un cal în poziţia `(x0,y0)`. Afişaţi cel mai scurt traseu pe care trebuie să-l parcurgă calul pentru a ajunge în poziţia `(x1,y1)`, fără a trece de două ori prin aceeaşi poziţie.

Exemplu. Pentru `n=5`, `x0=1`, `y0=1`, `x1=5`, `y1=5` soluţia este `(1,1),(2,3),(1,5),(3,4),(5,5)`.

135. Se dă un teren dreptunghiular cu 3 linii şi `2n` coloane. Se cere să se determine toate posibilităţile de pavare a acestui teren cu piese de domino, o piesă de domino putând fi plasată orizontal sau vertical pe teren.

Exemplu. Pentru `n=3` o soluţie este următoarea
```
1 2 3 3 4 4
1 2 5 6 6 7
9 9 5 8 8 7
```
în care numerele indică numărul de ordine al piesei folosite în acea poziţie.

136. Un soldat trebuie să parcurgă un teren minat pentru a ajunge în propriile linii. Să se determine cel mai scurt drum prin care soldatul ajunge nevătămat la camarazii săi. Se dau: poziţia iniţială a soldatului, poziţiile minelor, şi se ştie că propriile linii se află în afara terenului minat, de orice parte a sa. Deplasarea soldatului se poate face doar ortogonal.

Exemplu. Dacă terenul minat este codificat prin matricea

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix},$$

unde cu 1 am notat o zonă de teren minat, iar poziția inițială a soldatului este (2,2) atunci o soluție (nu unica) este (2,2), (3,2), (3,1), (3,0) (adică a ajuns în liniile proprii).

137. Pe o tablă de șah de dimensiune n se află un cal în poziția (x0,y0). Afișați toate posibilitățile calului de a ajunge în poziția (x1,y1), fără a trece de două ori prin aceeași poziție și știind că în anumite căsuțe, cu coordonatele citite de la tastatură calul nu poate sări, fiind inundate.

Exemplu. Dacă n=5, x0=1, y0=1, x1=5, y1=5 iar căsuțele inundate sunt: (2,3), (3,4), (5,2), (5,3) atunci câteva soluții sunt:

(1,1), (3,2), (1,3), (2,1), (3,3), (1,2), (2,4), (4,3), (5,5)

(1,1), (3,2), (2,4), (1,2), (3,1), (4,3), (5,5)

(1,1), (3,2), (4,4), (2,5), (3,3), (1,2), (3,1), (4,3), (5,5)

(1,1), (3,2), (5,1), (4,3), (5,5)

138. Pe o tablă de șah de dimensiune n se află un cal în poziția (x0,y0). Afișați cel mai scurt traseu pe care trebuie să-l parcurgă calul pentru a ajunge în poziția (x1,y1), fără a trece de două ori prin aceeași poziție și știind că în anumite căsuțe, cu coordonatele citite de la tastatură calul nu poate sări, fiind inundate.

Exemplu. Pentru aceleași date de intrare ca la problema anterioară soluția este (1,1), (3,2), (2,4), (4,3), (5,5).

139. Într-un labirint de dimensiune mxn se găsește un om în poziția (x,y). Să se găsească toate variantele de a ieși din labirint știind că în fiecare căsuță se găsește un număr cuprins între 0 și 15 codificând în binar dacă ușile spre celelalte celule (în ordinea N,V,S,E de la stânga la dreapta) sunt deschise sau nu. De exemplu numărul $13_{10}=1101_2$ precizează că ușile spre celulele din N, V și E sunt deschise iar cea spre S este închisă. Se știe de asemenea că omul nu are voie să treacă de mai multe ori prin aceeași cameră.

Exemplu. Dacă m=5, n=4, "harta" labirintului $\begin{pmatrix} 4 & 5 & 9 & 12 \\ 8 & 9 & 11 & 6 \\ 14 & 10 & 15 & 3 \\ 8 & 6 & 9 & 10 \\ 3 & 5 & 6 & 12 \end{pmatrix}$, iar poziția inițială a omului este (3,2) atunci o soluție (nu unica) este (3,2), (2,2), (2,3), (3,3), (4,3), (4,4), (5,4), (5,3), și ieșirea spre sud.

140. Pe un teren de dimensiune dreptunghiulară, cu denivelări, se află un sportiv care dorește să se antreneze pentru un concurs de alpinism. Cunoscând altitudinea fiecărei porțiuni din teren și poziția inițială a alpinistului, să se determine un traseu de lungime minimă pe care trebuie să-l parcurgă sportivul pentru a ajunge într-o poziție de altitudine maximă. Se știe că sportivul nu dorește să coboare deloc, dar poate merge pe loc drept. De asemenea el se poate deplasa ortogonal sau diagonal cu un singur pas.

141. Să se modifice programul anterior astfel încât în cazul în care există mai multe trasee de lungime minimă care respectă restricțiile indicate, se vor afișa toate.

Exemplu. Dacă harta terenului este $\begin{pmatrix} 16 & 2 & 12 & 20 & 13 \\ 4 & 5 & 9 & 3 & 15 \\ 8 & 13 & 16 & 19 & 20 \\ 14 & 5 & 7 & 7 & 9 \\ 18 & 3 & 16 & 8 & 20 \end{pmatrix}$ iar alpinistul se află inițial în punctul de coordonate (4,2) atunci soluțiile problemei sunt:

(4,2), (3,3), (3,4), (3,5)

(4,2), (4,3), (3,4), (3,5)

(4,2), (4,3), (4,4), (3,5)

(4,2), (4,4), (5,5)

(4,2), (4,3), (5,4), (5,5).

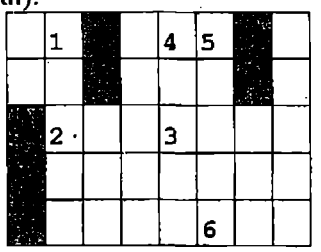
142. Modificați programul anterior astfel încât cunoscând altitudinea fiecărei porțiuni din teren, poziția inițială a alpinistului și poziția finală în care dorește să ajungă, să se determine un traseu pentru care suma diferențelor de altitudine între pozițiile consecutive de pe traseu să fie maximă. Nici de această dată alpinistul nostru nu are voie să coboare.

Exemplu. Pentru terenul dat de matricea $\begin{pmatrix} 16 & 2 & 12 & 20 \\ 4 & 5 & 9 & 30 \\ 8 & 13 & 16 & 19 \end{pmatrix}$, poziția inițială (1,2), iar cea finală (3,4), soluția este (1,2), (2,1), (2,2), (2,3), (3,2), (3,3), (3,4).

143. Gigel se află în poziția (x0,y0) a unui teren de dimensiune m×n. El poate efectua pași doar de anumite lungimi date de la tastatură, în una din direcțiile N, S, E, V, și nu poate păși într-o poziție mlăștinoasă pe teren. Pozițiile

mlăştinoase se citesc de la tastatură. Să se găsească toate modalităţile lui Gigel de ajunge în poziţia finală `(x1,y1)` în ipoteza că Gigel nu poate trece de două ori prin aceeaşi căsuţă.

Exemplu. Pentru `m=5`, `n=8`, lungimile posibile ale paşilor `(1,2,3,4,7)`, labirintul următor (căsuţele gri reprezintă zonele mlăştinoase) şi `(x0,y0)=(1,1)`, `(x1,y1)=(5,5)`, un posibil traseu este marcat cu cifre pe labirint (numărul pasului):



144. Un cal şi un rege se află pe o tablă de şah. Unele câmpuri sunt "arse", poziţiile lor fiind cunoscute. Calul nu poate călca pe câmpurile arse, iar orice mişcare a calului arde câmpul pe care se plasează. Să se afle o succesiune de mutări permise prin care calul să ajungă la rege şi să revină la poziţia iniţială. Poziţia iniţială a calului precum şi poziţia regelui sunt considerate nearse.

Exemplu. Pentru o tablă de dimensiune `5×5`, cu `5` poziţii arse şi anume `(1,4)`, `(2,1)`, `(2,2)`, `(2,5)`, `(4,5)`, poziţia iniţială a calului `(1,1)` iar cea a regelui `(5,5)`, atunci o soluţie este următoarea

```
13  8  -  -  3
 -  -  2  7  -
 - 12  9  4  -
10  -  6  -  -
 -  - 11  - 5
```

145. Edili oraşului Sibiu şi-au propus să îmbunătăţească aspectul zonei centrale a oraşului. Unul din proiecte este de a schimba dalele de pe "terasa" din faţa magazinului "Dumbrava". Se presupune că aceasta are forma dreptunghiulară ce poate fi împărţită în `m*n` căsuţe pătratice de aceeaşi dimensiune. Edilii au la dispoziţie dale de marmură având `4` culori diferite. Să se afişeze toate posibilităţile de a realiza pavarea respectându-se următoarele condiţii:

nu trebuie să existe două dale de aceeaşi culoare în căsuţe învecinate ortogonal (sus, jos, stânga, dreapta) sau diagonal;

fiecare dală va fi înconjurată de cel puţin câte o dală din toate celelalte trei culori.

Exemplu. Dacă dimensiunile terasei sunt 4×5 atunci o modalitate de pavoazare este $\begin{matrix} 1 & 2 & 4 & 2 & 1 \\ 4 & 3 & 1 & 3 & 4 \\ 1 & 2 & 4 & 2 & 1 \end{matrix}$, unde numerele indică culoarea dalei respective.

146. Se dau n^2 piese pătratice, care au pe fiecare latură numere întregi, reprezentând culori. Se cere să se plaseze piesele într-o matrice n×n. Două piese pot fi vecine în matrice dacă muchiile care le unesc au aceeaşi culoare. Culorile pieselor se citesc în ordinea sus, stânga, jos, dreapta. Nu este permisă rotirea pieselor.

Exemplu. Pentru n=4 şi piesele având următoarele culori:

```
Piesa 1: 1 2 4 3        Piesa 2: 5 3 7 6
Piesa 3: 4 8 10 9       Piesa 4: 7 9 12 11
```

pot fi plasate numai astfel $\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$.

147. Planul unui apartament este reprezentat sub forma unei matrice dreptunghice cu m linii şi n coloane. Pereţii apartamentului sunt reprezentaţi prin valoarea 1, uşile sunt reprezentate prin valoarea 2 dacă sunt închise sau 3 dacă sunt deschise. Într-o poziţie din apartament se află un bebeluş care ştie să meargă în patru labe, doar ortogonal, şi care vrea să "exploreze" apartamentul. Afişaţi planul apartamentului, în care marcaţi toate poziţiile în care poate ajunge bebeluşul ştiind că el este mult prea mic pentru a putea deschide uşile din apartament.

Exemplu. Dacă planul apartamentului este $\begin{matrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 3 & 0 & 0 & 0 & 0 & 1 \\ 1 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 3 & 1 \\ 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{matrix}$ iar poziţia iniţială a bebeluşului este (3,7) atunci zonele în care poate ajunge bebeluşul sunt marcate cu * pe următoarea matrice

```
1 1 1 1 1 1 1 1 1 1
1 * * * 1 * * * * 1
1 * * * 3 * * * * 1
1 2 2 1 1 1 1 1 3 1
1 0 0 0 0 2 * * * 1
1 1 1 1 1 1 1 1 1 1
```

148. O pajiște de formă dreptunghiulară este împărțită în `n×n` pătrate identice. În fiecare pătrat se găsește iarbă de o anumită culoare (culoarea este codificată printr-un număr între `1` și `10`). Pe acest teren se găsește o oaie **roz**. Știind că oița nu paște decât iarbă **albastră** (culoare cu codul `1`) să se determine o poziție de unde poate să înceapă oaia să pască și numărul de pătrățele în care oaia poate paște dacă se știe că:

oaia nu se poate deplasa decât orizontal sau vertical în una din căsuțele vecine;

nu poate călca pe iarbă de altă culoare decât cea albastră (se decolorează și moare);

oaia este lacomă și vrea să mănânce cât mai multă iarbă albastră.

Exemplu. Dacă "harta" terenului este
```
1 1 2 2 5
4 1 1 2 5
3 3 2 1 5
4 4 5 1 1
1 2 3 4 5
```
atunci evident porțiunea pe care va paște oaia este porțiunea albastră (cu cod `1`) din colțul stânga sus a terenului având dimensiunea de `4`.

149. Într-o cameră cu dale pătrate se află o maimuță și `nb` banane. Numărul bananelor, poziția (coordonatele) maimuței și pozițiile bananelor se citesc de la tastatură. Maimuța se poate deplasa cu câte o poziție în fiecare din cele `8` poziții vecine. Energia consumată este egală cu o unitate la deplasarea pe orizontală și verticală, respectiv `1.41` unități pentru deplasarea pe diagonală.

Se cere traseul optim și energia totală consumată de maimuță pentru a ajunge la toate bananele. Traseul optim este identificat printr-o energie consumată minimă.

Exemplu. Dacă dimensiunea camerei este `3×4`, `nb=4`, pozițiile bananelor sunt `(1,2)`, `(2,1)`, `(3,2)`, `(3,4)`, iar poziția inițială a maimuței `(1,4)`, atunci un traseu cerut este: `(1,4)`, `(1,3)`, `(1,2)`, `(2,1)`, `(3,2)`, `(3,3)`, `(3,4)`, iar energia consumată este `6.82`.

150. Se consideră un triunghi de numere în care pe prima linie apare un număr, pe a doua linie apar două numere etc. Scrieți un program care să calculeze cea mai mare dintre sumele numerelor ce apar pe drumurile ce pleacă din vârf și ajung la bază, astfel încât:

în fiecare drum succesorul unui număr se află pe rândul de mai jos și anume dedesubt sau pe diagonală la dreapta;

numărul de rânduri este strict pozitiv;

toate numerele ce apar în triunghi sunt întregi cuprinși între `0` și `99`.

Exemplu. Pentru triunghiul

```
7
3 8
8 1 0
2 7 4 4
4 5 2 6 5
```

valoarea căutată este 30.

151. Pe o masă de dimensiune n×n sunt aşezate n^2 prisme, fiecare având baza un pătrat de latura 1 şi diverse înălţimi. Pe una din prisme se aşează un melc care poate trece de pe o prismă pe alta (paralel cu marginile mesei) numai dacă prismele au aceeaşi înălţime sau dacă melcul coboară. Să se precizeze dacă şi pe ce traseu poate melcul să coboare de pe masă.

Exemplu. Dacă înălţimile fiecărei prisme sunt date în matricea $\begin{pmatrix} 1 & 6 & 9 & 4 \\ 2 & 3 & 10 & 24 \\ 54 & 6 & 9 & 78 \\ 4 & 5 & 47 & 12 \end{pmatrix}$ iar poziţia iniţială a melcului este (2,2) atunci singurele soluţii sunt:

(2,2), (2,1), (2,0)
(2,2), (2,1), (1,1), (1,0)
(2,2), (2,1), (1,1), (0,1).

152. Un anumit etaj al muzeului "Brukenthal" din Sibiu este reprezentat schematic sub forma unei matrice ce conţine caracterele 1 şi 0, unde 1 reprezentă "zid", iar "0" spaţiu liber. Spre exemplu, un etaj cu patru încăperi arată în felul următor:

```
1 1 1 1 1 1
1 1 0 1 0 1
1 1 0 1 1 1
1 1 1 0 0 1
1 0 1 1 1 1
1 1 1 1 1 1
```

Problema cere următoarele:

a) Determinaţi câte încăperi are etajul respectiv al muzeului;

b) Determinaţi care este încăperea cu suprafaţa cea mai mare;

c) ajutaţi directorul muzeului să aleagă un perete (o singură poziie de 1 din matrice) pe care să îl îndepărteze, pentru a obţine cea mai mare cameră posibilă, în scopul organizării unei expoziţii speciale.

Exemplu. Pentru harta de mai sus cu se găsesc 4 camere, cea mai mare cameră are suprafaţa egală cu 2, iar zidul dărâmat este cel de la coordonatele (3,4). Harta după dărâmarea acestui zid este:

```
1 1 1 1 1 1
1 1 0 1 0 1
1 1 0 0 1 1
1 1 1 0 0 1
1 0 1 1 1 1
1 1 1 1 1 1
```

iar camera astfel obținută are suprafața egală cu 5.

153. Se dă un careu de dimensiune `m×n` cu litere și un cuvânt. Să se găsească prefixul de lungime maximă al cuvântului în careu. Două litere consecutive ale prefixului se vor afla în careu pe două poziții consecutive ale aceleiași linii sau coloane.

Exemplu. Pentru careul

```
e f s p
c c a u
o b c t
v n a s
```

și cuvântul `castelan`, prefixul de lungime maximă este `cast` poziționat astfel: `(3,3)`, `(4,3)`, `(4,4)`, `(3,4)`.

154. Scufița Roșie pleacă la bunica sa prin pădure. Pădurea este de formă dreptunghiulară de dimensiune `m×n`. Casa Scufiței Roșii se află în pătrățelul de coordonate `(1,1)` iar casa bunicii în pătrățelul de coordonate `(m,n)`. Fiecare pătrățel al pădurii conține o floare, un copac sau iarbă. În pătratul de coordonate `(x,y)` al pădurii se află lupul. Dorința Scufiței Roșii este să ajungă la bunica sa astfel încât să culeagă cât mai multe flori în drumul său, flori care să-i aducă o bucurie în plus bunicii sale. Bineînțeles Scufița trebuie să evite să treacă prin apropierea lupului (să nu treacă prin nici una din căsuțele vecine pe linie, coloană sau diagonală cu căsuța lupului). Deplasarea Scufiței se poate face în oricare din direcțiile `N`, `S`, `E`, `V`, cu câte un pătrățel la fiecare pas, fără a putea trece prin pătrățelele care conțin copaci. Puteți să o ajutați pe Scufiță să-și îndeplinească dorința?

Exemplu. Pentru `m=6`, `n=7`, numărul de flori `14`, numărul de copaci `10`, coordonatele florilor `(1,5)`, `(2,2)`, `(2,3)`, `(2,6)`, `(3,1)`, `(3,4)`, `(3,6)`, `(3,7)`, `(4,3)`, `(4,5)`, `(5,2)`, `(5,3)`, `(5,4)`, `(6,5)`, coordonatele copacilor `(1,3)`, `(1,4)`, `(1,6)`, `(1,7)`, `(2,5)`, `(3,2)`, `(4,2)`, `(5,5)`, `(5,6)`, `(6,3)`, coordonatele lupului `(3,5)`, se obține următoarea soluție: `(1,1)`, `(2,1)`, `(2,2)`, `(2,3)`, `(3,3)`, `(4,3)`, `(5,3)`, `(5,4)`, `(6,4)`, `(6,5)`, `(6,6)`, `(6,7)`.

155. Se dau numerele naturale `n` și `k`, `0<=k<=n<=10`. Se consideră rețeaua de puncte `P` din planul `xOy`: `P(p,q)`, `p,q∈N`, `p∈{0,1,...,k}` și `q∈{0,1,...,n-k}`. Să se determine toate drumurile minimale din rețea care

leagă punctele O(0,0) şi M(k,n-k). Un drum în reţea leagă două punte oarecare ale reţelei şi este format din laturi ale pătratelor reţelei. Afişarea unui drum presupune afişarea coordonatelor punctelor care formează drumul.

Exemplu. Pentru n=3 şi k=1 soluţiile sunt:

(0,0), (1,0), (1,1), (1,2)
(0,0), (0,1), (1,1), (1,2)
(0,0), (0,1), (0,2), (1,2).

156. Fie nxm eprubete aşezate într-un stativ de formă dreptunghiulară, cu n linii a câte m eprubete fiecare. În fiecare eprubetă se găseşte o substanţă chimică (multe eprubete pot conţine aceeaşi substanţă). O persoană efectuează nişte experimente folosindu-se de substanţele date. Se ştie că persoana respectivă a efectuat experienţele în aşa fel încât la fiecare reacţie a folosit doar două substanţe şi, în plus, aceste substanţe se găseau în eprubete învecinate pe orizontală sau pe verticală. Substanţele aflate în stativul dat sunt astfel alese încât oricare două dintre ele, dacă vor reacţiona, vor da naştere la exact doi produşi de reacţie. Reacţiile se citesc de la intrare sub forma i, j, k, l având următoarea semnificaţie: dacă substanţa i reacţionează cu substanţa j, atunci vor rezulta substanţele k şi l. După reacţie, substanţa k va fi pusă în locul substanţei i, iar substanţa l în locul substanţei j. Cunoscând modul în care erau amplasate substanţele în configuraţia iniţială şi cea finală, se cere să se reconstituie etapele necesare trecerii de la configuraţia iniţială la cea finală.

157. Fiind dat rombul:

```
      A
    L   L
  G   G   G
O   O   O   O
  R   R   R   R   R
  I   I   I   I
    T   T   T
      M   M
        I
```

să se afişeze sub forma în care se găsesc în romb, toate cuvintele ALGORITMI ale căror litere succesive apar în romb pe poziţii succesive.

Exemplu. O soluţie este următoarea (afişarea se va face în acest mod!):

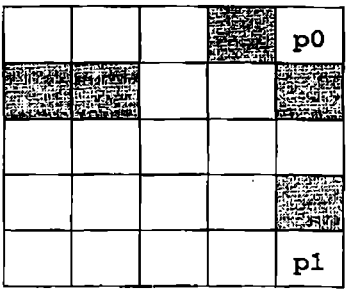
```
    A
  L
    G
      O
    R
        I
          T
    M
```

158. Pe o tablă de şah de dimensiuni MxN anumite pătrate sunt ocupate. Dându-se un pătrat iniţial şi unul final, se cer următoarele:

1) să se determine dacă se poate ajunge cu un nebun din pătratul iniţial în cel final, fără a trece prin pătrate ocupate;

2) dacă răspunsul la 1) este afirmativ, se cere să se afişeze numărul minim de mutări prin care se poate ajunge din pătratul iniţial în cel final.

Exemplu. Pentru următoarea tablă



în care am marcat cu gri căsuţele ocupate iar cu `p0` şi `p1` poziţia iniţială respectiv finală a nebunului, atunci soluţia este `(1,5),(2,4),(3,5),(4,4),(5,5)`.

159. Fie `A` un tablou de dimensiune `nxn` (`n>2`), având elemente din mulţimea `{0,1,...,9}` citite de la tastatură. Se cere ca pornind dintr-o poziţie convenabil aleasă şi deplasându-se în orizontal, vertical sau diagonal cu o căsuţă (în limita tabloului) să se parcurgă un traseu care să fie marcat cu cifrele unui număr natural `p` dat de la tastatură. Să se găsească toate traseele posibile care satisfac condiţiile enunţate mai sus în următoarele cazuri:

a) se respectă ordinea cifrelor numărului `p`;

b) nu se respectă această ordine.

Exemplu. Pentru tabloul $\begin{pmatrix} 1 & 6 & 6 & 4 & 2 \\ 5 & 2 & 8 & 9 & 6 \\ 2 & 4 & 3 & 4 & 2 \\ 6 & 4 & 8 & 3 & 5 \end{pmatrix}$ de dimensiune `4x5`, şi numărul `p=264` se obţin:

a) 7 soluţii printre care

`(2,2),(1,3),(1,4)`
`(1,5),(2,5),(1,4)`
`(3,5),(2,5),(1,4)`
`(3,1),(4,1),(3,2)`
`(1,5),(2,5),(3,4)`
`(3,5),(2,5),(3,4)`
`(3,1),(4,1),(4,2)`

a) cele 7 soluții anterioare la care se adaugă și alte soluții ca de exemplu:
(1,2),(2,2),(3,2)
(1,3),(1,4),(1,5) etc.

160. Pe o tablă de dimensiune **m n** se consideră o configurație de maxim **m n-1** pietre. Aceste pietre se pot muta pe orizontală sau verticală doar peste un vecin, care în urma săriturii va dispărea. Pentru piatra de pe poziția (i,j) vecinii săi sunt (i-1,j), (i+1,j), (i,j-1), (i,j+1). Saltul se face din căsuța (i,j) în căsuțele (i-2,j), (i+2,j), (i,j-2), (i,j+2) dacă aceasta este liberă, s-a sărit peste un vecin și nu s-a ieșit din tabla de joc. Să se determine dacă pentru o configurație dată se poate ajunge la o singură piatră pe tablă, în caz afirmativ să se furnizeze cel puțin o soluție corectă.

Exemplu. Pentru configurația:
```
0 1 0 0
1 0 0 0
0 1 1 0
```
o soluție este:

(3,3) (3,1) dispare (3,2)
(3,1) (1,1) dispare (2,1)
(1,1) (1,3) dispare (2,2).

161. Se dă o rețea dreptunghiulară de dimensiune **m n**. În anumite noduri, ale căror coordonate se citesc de la tastatură, se află morcovi, în cantități precizate. Un iepuraș se poate deplasa orizontal sau vertical, pornind dintr-un punct ce conține morcovi. Deplasarea între două noduri alăturate presupune consumarea unui morcov. Trecerea printr-un nod cu morcovi presupune câștigarea întregii cantități de morcovi din nod. Să se stabilească un nod de plecare și un traseu de deplasare astfel încât:

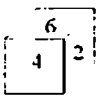
iepurașul să culeagă toți morcovii;

numărul final de morcovi rămași să fie maximă.

Inițial iepurașul nu are nici un morcov și nu se poate deplasa fără morcovi. Dacă nu este posibilă culegerea tuturor morcovilor se va afișa "Iepurașul moare"

Exemplu. Dacă există 5 căsuțe cu morcovi și anume
```
5 2
3
5 2
```
atunci iepurașul va pleca din punctul (1,1) și va parcurge următorul traseu (1,1), (1,2), (1,3), (1,4), (2,4), (3,4), (3,5), (4,5), (5,5), (5,4), (5,3), (5,2).

punctelor de pe oricare două fețe opuse este 7. Să se afișeze toate posibilitățile de a aduce zarul în colțul din dreapta jos prin răsturnare pe una din fețele învecinate celei pe care stă, fără a trece de mai multe ori prin aceeași poziție și ocolind obstacolele existente pe tablă. Dimensiunea zarului coincide cu dimensiunea unui careu al tablei. În soluții se va afișa numai valorile situate pe fața de sus a zarului.



Exemplu. Pentru m=4 și n=7 și careul $\begin{pmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$ o soluție este $\begin{pmatrix} 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 5 & 6 & 2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 5 & 3 \end{pmatrix}$.

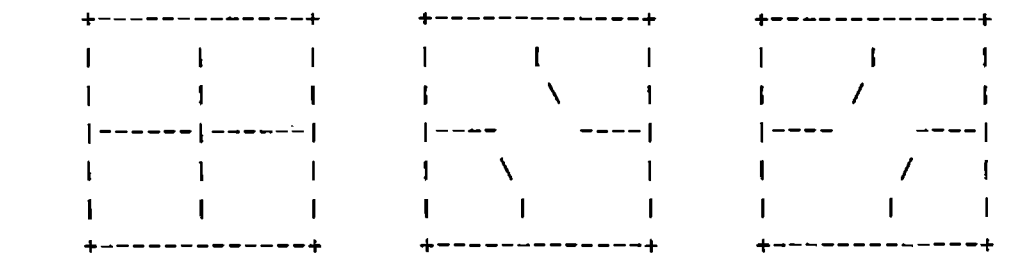
163. În colțul din stânga sus al unei table de șah de dimensiune m×n, se găsește un zar aflat în poziția din figura de la problema anterioară. Se știe că zarul are proprietatea că suma punctelor de pe oricare două fețe opuse este 7. În fiecare căsuță a tablei se găsește un număr cuprins între 1 și 6. Căsuța din colțul stânga sus are înscrisă valoarea 1. Se cere să se găsească cel mai scurt traseu de aducere a zarului în colțul din dreapta jos prin răsturnare pe una din fețele învecinate celei pe care stă, fără a trece de mai multe ori prin aceeași poziție. Răsturnarea pe una din fețe se poate face doar dacă numărul înscris în căsuța peste care se face răsturnarea va coincide cu valoarea înscrisă pe partea de jos a zarului. La afișarea soluției se va afișa numai valorile situate pe fața de sus a zarului.

Exemplu. Pentru tabla $\begin{pmatrix} 1 & 2 & 6 & 5 & 3 & 6 \\ 4 & 2 & 3 & 4 & 1 & 2 \\ 3 & 2 & 6 & 6 & 2 & 1 \\ 2 & 4 & 4 & 5 & 6 & 3 \\ 2 & 5 & 6 & 2 & 1 & 5 \end{pmatrix}$ se va afișa `6,5,1,2,3,6,5,1,4,2.` care corespunde următorului traseu: `(1,1)->(1,2)->(1,3)->(1,4)->(2,4)->(2,5)->(3,5)->(4,5)->(4,6)->(5,6).`

164. Într-un acvariu de forma unui paralelipiped dreptunghic de dimensiune $n_1 \times n_2 \times n_3$ se află un căluț de mare care se poate deplasa ca un cal de șah în oricare din cele trei planuri ortogonale. În anumite puncte ale acvariului se găsesc pești prădători. Căluțul dorește să ajungă din poziția sa curentă

(x0,y0,z0) într-o poziție (x_1, x_2, x_3) unde se găsește hrană, însă nu dorește să treacă prin căsuțele ocupate de pești. Ajutați-l pe căluț să găsească cel mai scurt drum posibil.

165. Se consideră o tablă de n*n pătrățele și un număr de n^2 piese pătrate, fiecare piesă putând avea una din formele:



Piesa 1 Piesa 2 Piesa 3

Se observă că piesa 1 are conectate marginile N-S și E-V, piesa 2 are conectate marginile N-E și S-V, iar piesa 3 are conectate marginile N-V și S-E. Subliniem că cele două linii din piesa 1 nu se intersectează, ci trec una pe sub cealaltă.

Se cere să se aşeze cele n^2 piese pe tablă în aşa fel încât să se obțină un drum care:

să treacă prin fiecare pătrat exact o dată;

să nu se autointersecteze;

să pornească din colțul de NV al tablei (linia 1, coloana 1), începând de la exteriorul tablei (fie din nord, fie din vest);

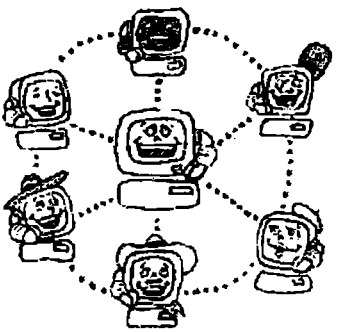
să se termine în colțul de SE al tablei (linia n, coloana n) și să părăsească tabla.

Se citește de la tastatură: numărul N1, N2, N3, al pieselor din tipurile 1, 2 și 3 și dimensiunea n a tablei. Soluția se va memora în fișierul text "PIESE.TXT" sub forma unei matrice cu n*n numere separate prin spații, reprezentând tipul piesei din fiecare pătrățel. Dacă există mai multe soluții se va memora doar una singură. Dacă nu există nici o soluție fișierul "PIESE.TXT" va conține mesajul "IMPOSIBIL"

Exemplu. Pentru N1=6, N2=9, N3=10, N=25 o soluție este

```
1  2  3  1  2
3  3  1  3  3
1  3  3  2  2
1  2  2  3  3
2  1  3  2  2
```

7.1. Metoda "Divide et Impera"



166. Fie n un număr natural nenul şi a_1, a_2, , a_n o secvenţă de n numere naturale citite de la tastatură. Scrieţi un program, bazat pe metoda "Divide et Impera" care să calculeze cmmdc(a_1,a_2,...,a_n).

Exemplu. cmmdc(1540,6076,490)=28.

167. Să se scrie un program pentru determinarea minimului dintr-un şir de numere întregi folosind metoda "Divide et Impera"

168. Fie P(x) şi Q(x) două polinoame de grad n, cu coeficienţi reali daţi sub forma a doi vectori. Să se înmulţească cele două polinoame folosind metoda "Divide et Impera"

169. Fie n numere întregi a_1, a_2,..., a_n. Să se determine (folosind un algoritm "Divide et Impera") cel de al k-lea cel mai mic element din şir.

170. Se dă un vector cu n numere întregi. Se împarte tabloul în trei grupe (cât mai echilibrate ca dimensiune), eliminându-se în fiecare grup mijlocul sau mijloacele (dacă în acel grup există un număr par de elemente). Se repetă procedeul pentru fiecare din cei şase subvectori obţinuţi (din fiecare din cele trei grupe se obţin după eliminarea mijlocului/mijloacelor doi subvectori). Procedeul se repetă de p ori. Să se afişeze vectorul obţinut după terminarea algoritmului.

Exemplu. Pentru n=50 şi x[i]=i, i=1,2,...,n, p=2 se va obţine în final vectorul x=(5,7,14,16,19,21,22,24,28,30,31,33,36,38,39,41, 45,48,50).

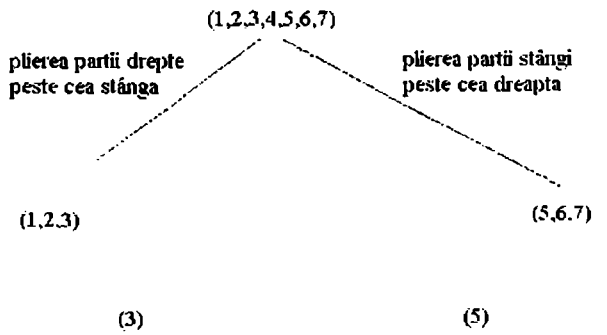
171. Să se generalizeze problema anterioară pentru k grupe.

172. Generalizaţi problema turnurilor din Hanoi pentru două tije de manevră. Deci enunţul sună astfel: Se dau 4 tije numerotate 1,2,3,4, şi n discuri de diametre diferite. Iniţial, toate discurile sunt plasate pe tija 1 în ordinea descrescătoare a diametrelor, considerând sensul de la bază la vârf. Problema cere să se mute discurile de pe tija 1 pe tija 2, folosind ca tije de manevră tijele 3 şi 4 şi respectând regulile: la fiecare pas se mută un singur disc, un disc poate fi aşezat peste un alt disc mai mare sau poate fi primul disc de la baza tijei.

173. Se consideră un vector de lungime n. Definim plierea vectorului prin suprapunerea unei jumătăţi (numită donatoare) peste cealaltă jumătate (numită receptoare). Dacă n este impar, elementul din mijloc este eliminat. În

urma plierii se obține tot un vector ale cărui elemente încep de la o anumită poziție p și se termină la o poziție u. Plierea continuă în mod repetat până se ajunge la un vector de lungime 1 (p=u), numit element final. Se cere să se afișeze toate elementele finale posibile.

Exemplu. Pentru n=7 elementele finale sunt x_1, x_3, x_5 și x_7 după cum rezultă din schema următoare.



174. Scrieți un program care calculează valoarea unui polinom într-un punct folosind metoda "Divide et Impera"

175. Fie un tablou neordonat de n numere întregi. Să se determine prin metoda "Divide et Impera" toate pozițiile pe care apare în tablou o anumită valoare v.

176. Se dau n numere întregi pentru care se cere să se afișeze modalitatea de calcul a produsului lor prin metoda "Divide et Impera"

Exemplu. Pentru n=5 se va afișa $((x_1*x_2)*x_3)*(x_4*x_5)$.

177. Să se scrie un program recursiv de înmulțire a două matrice A (cu m linii și n coloane) și B (cu n linii și p coloane) observând că aceasta se poate face astfel:

$$AB = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix}\begin{pmatrix} B_1 & B_2 \\ B_3 & B_4 \end{pmatrix} = \begin{pmatrix} A_1B_1 + A_2B_3 & A_1B_2 + A_2B_4 \\ A_3B_1 + A_4B_3 & A_3B_2 + A_4B_4 \end{pmatrix}$$

unde A_1, A_2, A_3, A_4, respectiv B_1, B_2, B_3, B_4, sunt submatrice ale matricei A, respectiv B, astfel:

$$A_1 = \begin{pmatrix} a_{11} & a_{12} & a_{1q} \\ a_{21} & a_{22} & a_{2q} \\ & & \\ a_{r1} & a_{r2} & a_{rq} \end{pmatrix} \quad A_2 = \begin{pmatrix} a_{1q+1} & a_{1q+2} & a_{1n} \\ a_{2q+1} & a_{2q+2} & a_{2n} \\ & \vdots & \\ a_{rq+1} & a_{rq+2} & a_{rn} \end{pmatrix}$$

$$A_3 = \begin{pmatrix} a_{r+11} & a_{r+12} & a_{r+1q} \\ a_{r+21} & a_{r+22} & a_{r+2q} \\ & & \\ a_{m1} & a_{m2} & a_{mq} \end{pmatrix} \quad A_4 = \begin{pmatrix} a_{r+1q+1} & a_{r+1q+2} & a_{r+1n} \\ a_{r+2q+1} & a_{r+2q+2} & a_{r+2n} \\ & & \\ a_{mq+1} & a_{mq+2} & a_{mn} \end{pmatrix}$$

analog pentru B, unde am notat $q = \left[\frac{n}{2}\right]$, $r = \left[\frac{m}{2}\right]$.

7.2. Metoda Greedy

178. O stație de servire trebuie să satisfacă cererile a n clienți. Timpul de servire necesar clientului i este ti, Se cere să se minimizeze timpul total de așteptare pentru clienții i, i=1,2,...,n.

179. Se consideră două mulțimi de numere întregi A={a1,...,am} și B={b1,...,bn}, n>=m. Să se determine o submulțime X={x1,x2,...,xm} a lui B astfel încât să se maximizeze valoarea expresiei $a_1x_1+a_2x_2+...+a_mx_m$.

180. Fiind date matricele $A_1, A_2, ..., A_n$ cu elemente numere reale unde pentru fiecare i=1,2,...,n matricea A_i are d_i linii și d_{i+1} coloane, se cere să se calculeze produsul $A_1 \times A_2 \times ... \times A_n$ efectuând cât mai puține înmulțiri de numere reale.

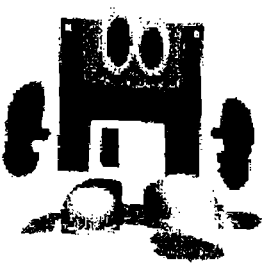
Exemplu. Pentru n=6 matrice având dimensiunile d=(30,35,15,5,10,20,25) o înmulțire optimă se obține conform următoarei parantezări: $(A_1(A_2A_3))((A_4A_5)A_6)$ numărul total de înmulțiri cu numere reale fiind 15125.

181. Se dă o matrice A cu m linii și n+1 coloane. Fiecare linie a matricei conține un număr de a[i,0] elemente ordonate crescător și anume a[i,1], a[i,2], a[i,3], ..., a[i,a[i,0]]. Se cere să se găsească o modalitate optimă (cu un număr cât mai mic de comparații) de a interclasa aceste șiruri, rezultatul fiind memorat într-un vector b.

182. Un grup este format din n persoane, unde n este număr natural (1<=n<=100). Fiecare persoană poate avea cel mult 3 dușmani. Se cere, dacă este posibil, să împărțiți cele n persoane în două grupuri, astfel încât, în fiecare grup, o persoană are cel mult un dușman. Se știe că dacă x este dușmanul lui y atunci și y este dușmanul lui x.

183. La festivalul internațional de teatru de la Sibiu sunt programate într-o zi n spectacole. Pentru fiecare spectacol se cunoaște ora de începere a spectacolului precum și durata acestuia. Un pasionat al teatrului își propune să vizioneze cât mai multe din aceste spectacole, dar bineînțeles că nu poate vedea două spectacole care se desfășoară simultan. Dați-i o mână de ajutor. Soluția obținută cu metoda Greedy este întotdeauna soluția optimă?

7.3. Metoda programării dinamice



184. Fiind date matricele A_1, A_2, ..., A_n cu elemente numere reale unde pentru fiecare $i=1,2,...,n$ matricea A_i are d_i linii şi d_{i+1} coloane, se cere să se determine o ordine de efectuare a produsului $A_1 \times A_2 \times ... \times A_n$ astfel încât să se efectueze cât mai puţine înmulţiri de numere reale.

Exemplu. Pentru $n=6$ matrice având dimensiunile $d=(30,35,15,5,10,20,25)$ o înmulţire optimă se obţine conform următoarei parantezări: $(A_1(A_2A_3))((A_4A_5)A_6)$ numărul total de înmulţiri cu numere reale fiind 15125.

185. Pe produsul cartezian $N \times N$ se defineşte operaţia:

$$(a,b)(b,c) = (a,c) \qquad \forall\ a,b,c \in N$$

despre care ştim că:

1) este asociativă

$$[(a,b)(b,c)](c,d) = (a,b)[(b,c)(c,d)];$$

2) efectuarea ei necesită $a*b*c$ secunde.

Fiind date $x_1, x_2, ..., x_n \in N$, $n \geq 3$, numere întregi, se cere să se determine timpul minim şi cel maxim în care se poate efectua

$$(x_1,x_2)(x_2,x_3)...(x_{n-1},x_n).$$

Să se afişeze de asemenea câte o modalitate de efectuare a acestei operaţii pentru a se obţine aceşti timpi de execuţie.

Exemplu. Pentru (16,9)(9,13)(13,1)(1,1)(1,14)(14,11)(11,3) se obţine timpul minim 499 pentru

[(16,9)[(9,13)(13,1)]][(1,1)[[(1,14)(14,11)](11,3)]

respectiv timpul maxim 7972 pentru

[[[(16,9)(9,13)][(13,1)[(1,1)(1,14)]]](14,11)](11,3).

186. Se citesc de la tastatură un şir de n numere naturale. Să se scrie un program care afişează cea mai lungă secvenţă şi cel mai lung subşir al şirul dat, cu proprietatea că suma elementelor sale este divizibilă cu 3.

Exemplu. Pentru şirul (5,8,3,12,4,1,2,8,6,4,2) o secvenţă de lungime maximă este (8,3,12,4,1,2,8,6,4). Cel mai lung subşir al şirului (2,3,5,7,9,0,14) este (2,3,5,9,0,14).

187. Un număr de `n` elevi sunt aşezaţi la ora de sport toţi pe un rând în ordinea crescătoare a înălţimii. Pentru exerciţiul următor, profesorul de sport doreşte să elimine din şir un număr minim de elevi astfel încât diferenţa de înălţime dintre oricare doi elevi vecini din şirul final să fie cel puţin `x`. Se ştie de asemenea că primul elev din şir nu trebuie eliminat.

Exemplu. Dacă `n=5`, `x=30` şi înălţimile elevilor sunt `(110,130,140,160, 190)` atunci o soluţie este selectarea elevilor `1,3,5`.

188. La festivalul internaţional de teatru de la Sibiu sunt programate într-o zi `n` spectacole. Pentru fiecare spectacol se cunoaşte ora de începere a spectacolului precum şi durata acestuia. Un pasionat al teatrului îşi propune să vizioneze cât mai multe din aceste spectacole, dar bineînţeles că nu poate vedea două spectacole care se desfăşoară simultan. Daţi-i o mână de ajutor.

Exemplu. Dacă `n=3` iar orele de început a spectacolelor respectiv durata lor sunt: `(10,4), (12,4), (13,2), (15,2), (16,2), (17,2)` atunci soluţia este vizionarea spectacolelor `1,4,6`.

189. Se dă un text format dintr-o succesiune de `n` cuvinte, de lungimi l_1, $l_2, \ldots, l_n$. Se doreşte tipărirea "uniformă" a paragrafului pe un anumit număr de linii, fiecare linie având cel mult `m` caractere. Criteriul de "uniformitate" se referă la minimizarea spaţiului suplimentar ce trebuie adăugat la sfârşitul fiecărei linii pentru a ajunge la `m` caractere pe linie.

Exemplu. Pentru textul "`Virginia Satir descrie comunicarea ca pe o umbrela uriasa care acopera si influenteaza tot ceea ce se intampla intre fiintele umane`" şi `m=20` o tipărire uniformă este

```
Virginia Satir
descrie comunicarea
ca pe o umbrela
uriasa care acopera
si influenteaza
tot ceea ce se
intampla intre
fiintele umane
```

190. Se consideră un şir de piese de domino. Fiecare piesă poate fi rotită în jurul centrului ei cu `180°`. Să se determine o `secvenţă` (adică subşir de elemente consecutive) de piese, de lungime maximă, în care oricare două piese alăturate au înscrise pe ele acelaşi număr: al doilea număr de pe prima piesă coincide cu primul număr de pe cea de a doua piesă.

Exemplu. Pentru secvenţa de piese `(1,2), (3,4), (5,1), (2,5), (6,2), (2,3)`, secvenţa cerută are lungimea `3` şi începe cu piesa de pe poziţia `3`: `(1,5), (5,2), (2,6)`.

191. Se consideră un şir de piese de domino. Fiecare piesă poate fi rotită în jurul centrului ei cu 180°. Să se determine un subşir (adică elemente nu neapărat consecutive în şir) de piese de lungime maximă în care oricare două piese alăturate au înscrise pe ele acelaşi număr: al doilea număr de pe prima piesă coincide cu primul număr de pe cea de a doua piesă.

Exemplu. Pentru secvenţa de piese (3,4), (1,4), (5,4), (3,2), (2,5), (3,5), (6,2) un subşir de lungime maximă este (1,4), (4,5), (5,2), (2,6).

192. Să se scrie numărul N ca sumă de numere întregi pozitive al căror produs este maxim posibil.

Exemplu. Pentru n=10 soluţia este (2,2,3,3).

193. Câte cuvinte de lungime N se pot forma cu litere din alfabetul {a,b,c,d} astfel încât a şi b să nu se afle pe poziţii alăturate ?

194. Se dă o scară cu n trepte (n<=100). Ştiind că un om poate să urce cel mult 3 trepte odată, să se afle toate posibilităţile de a urca scara cu efort minim. Efortul pentru urcarea unei trepte este k, efortul necesar pentru urcarea a două trepte este $k + \frac{1}{7}k$, iar pentru urcarea a trei trepte odată este $k + \frac{1}{3}k$.

Exemplu. Pentru n=11 şi k=5 soluţia este (3,3,2,2,1).

195. Date fiind numerele N şi S, se cere să se determine câte numere naturale de N cifre cu suma cifrelor S există.

Exemplu. Pentru N=6 şi S=15 se obţin 10746 numere.

196. Se dau două şiruri $X=(x_1,\ldots,x_m)$ şi $Y=(y_1,\ldots,y_n)$ şi se cere determinarea unui subşir comun de lungime maximă pentru X şi Y. Prin subşir al unui şir vom înţelege o secvenţă de elemente ale şirului, nu neapărat adiacente.

Exemplu. Dacă şirurile sunt: X='ABCBDAB' iar Y='BDCABA' atunci subşirul de lungime maximă este 'BCBA'

197. Se dau două numere naturale A şi B si un vector v cu n numere naturale ordonate crescător (n,A,B,v[i]<=100). Se cere să se afle dacă se poate trece din numărul A în numărul B, folosind următoarele operaţii:

a) adunarea la A a unui număr din v;

b) scăderea din A a unui număr din v.

În caz afirmativ se cere să se genereze un astfel de şir de operaţii de lungime minimă.

Exemplu. Pentru A=15, B=29 n=4, v=(3,10,12,30) atunci B se obţine din A prin următorul şir minimal de operaţii 29=15-10+12+12.

198. Se citesc n implicații de forma p_i->p_k, unde p_i și p_k sunt propoziții. Interpretarea este următoarea: dacă propoziția p_i este adevărată, atunci p_k este adevărată. Se citesc două propoziții x și y, aflate în mulțimea celor care apar în cadrul celor n reguli. Să se verifice dacă are loc implicația x->y și, în caz afirmativ, să se precizeze un număr minim de reguli care conduc de la propoziția x la propoziția y.

Exemplu. Dacă implicațiile sunt 1->2,1->6,2->5,3->4,4->2,4->5, 5->3, atunci implicația 2->6 nu poate avea loc, iar implicația 1->3 este realizabilă din trei reguli: 1->2, 2->5, 5->3.

199. Se dau două șiruri de numere întregi. Să se afișeze subșirul crescător de lungime maximă al șirului obținut prin intercalarea, în orice mod, a șirurilor date.

Exemplu. Pentru șirurile (3,1,4), (2,2,1,5) se obține (1,2,2,4,5).

200. În armată, o companie este alcătuită din n soldați. La inspecția de dimineață soldații stau aliniați în linie dreaptă în fața căpitanului. Acesta nu e mulțumit de ceea ce vede. E drept că soldații sunt așezați în ordinea numerelor de cod 1,2,..,n din registru, dar nu în ordinea înălțimii. Căpitanul cere câtorva soldați să iasă din rând, astfel ca cei rămași, fără a-și schimba locurile, doar apropiindu-se unul de altul (pentru a nu rămâne spații mari între ei) să formeze un șir în care fiecare soldat vede privind de-a lungul șirului, cel puțin una din extremități (stânga sau dreapta). Un soldat vede o extremitate dacă între el și capătul respectiv nu există un alt soldat cu înălțimea mai mare sau egală ca a lui.

Scrieți un program care determină, cunoscând înălțimea fiecărui soldat, numărul minim de soldați care trebuie să părăsească formația astfel ca șirul rămas să îndeplinească condiția din enunț.

Pe prima linie a fișierului de intrare ALINIERE.IN este scris numărul n (2<=n<=1000) al soldaților din șir, iar pe linia următoare un șir de n numere reale din intervalul [0.5,2.5], cu maximum 5 zecimale fiecare și separate prin spații. Al k-lea număr de pe această linie reprezintă înălțimea soldatului cu codul k (1≤k≤n).

Fișierul ALINIERE.OUT va conține pe prima linie numărul soldaților care trebuie să părăsească formația, iar pe linia următoare codurile acestora în ordine crescătoare, separate două câte două printr-un spațiu. Dacă există mai multe soluții posibile, se va scrie una singură.

Exemplu.

ALINIERE.IN	ALINIERE.OUT
8	4
1.86 1.86 1.30621 2 1.4 1 1.97 2.2	1 3 7 8

201. O persoană are la dispoziție un rucsac cu capacitate de G unități de greutate și intenționează să efectueze un transport în urma căruia să obțină un câștig.

Persoana are la dispoziție n tipuri de obiecte. Fiecare tip obiect se găsește în număr nelimitat. Pentru fiecare obiect se cunoaște greutatea și câștigul obținut în urma transportului său. Ce obiecte trebuie să aleagă persoana pentru a-și maximiza câștigul și care este acest câștig?

Exemplu. Pentru n=3 obiecte având greutățile (3,4,3) și valorile (2,3,8) soluția este de a lua două obiecte de tipul 3 valoarea totală fiind 16.

202. Se citește n număr natural. Se citesc apoi n numere naturale cu semnificația de valori ale unor bancnote. Se cere să se plătească (dacă este posibil) suma S utilizând valorile bancnotelor citite, utilizând un număr minim de bancnote. Se presupune că din fiecare tip se dispune de un număr nelimitat de bancnote.

Exemplu. Pentru 4 tipuri de monezi de valori 60, 59, 6, 1 și suma de plată S=124 se obține soluția: 2 monezi de 59, 1 monedă de 6.

203. Gigel are o "balanță" mai ciudată pe care vrea să o echilibreze. De fapt, aparatul este diferit de orice balanță pe care ați văzut-o până acum. Balanța lui Gigel dispune de două brațe de greutate neglijabilă și lungime 15 fiecare. Din loc in loc, la aceste brațe sunt atașate cârlige, pe care Gigel poate atârna greutăți din colecția sa de G greutăți (1<=G<=20) de valori distincte (numere naturale intre 1 si 25). Gigel poate atârna oricâte greutăți de orice cârlig, dar trebuie să folosească toate greutățile de care dispune.

Gigel a reușit să echilibreze relativ repede balanța dar acum dorește să afle în câte moduri poate fi echilibrată balanța.

Cunoscând amplasamentul cârligelor și setul de greutăți pe care Gigel îl are la dispoziție, scrieți un program care calculează în câte moduri se poate echilibra balanța.

Se presupune că este posibil să se echilibreze balanța. Balanța se echilibrează dacă suma produselor dintre greutăți și coordonatele unde ele sunt plasate este 0 (suma momentelor greutăților față de centrul balanței este 0).

Exemplu. Pentru 2 cârlige de coordonate -2 și 3 și 4 greutăți: 3,4,5,8 există 2 variante de echilibrare.

204. Pe o tablă de șah de dimensiune n se află un cal în poziția (x0,y0). Afișați cel mai scurt traseu pe care trebuie să-l parcurgă calul pentru a ajunge în poziția (x1,y1), fără a trece de două ori prin aceeași poziție.

Exemplu. Dacă n=8, (x0,y0)=(1,1), (x1,y1)=(8,7) atunci cel mai scurt drum este de lungime 5, iar un traseu posibil este (1,1),(3,2),(2,4), (4,5),(6,6),(8,7).

205. Se consideră un triunghi de numere în care pe prima linie apare un număr, pe a doua linie apar două numere etc. Scrieți un program care să calculeze cea mai mare dintre sumele numerelor ce apar pe drumurile ce pleacă din vârf și ajung la bază, astfel încât:

în fiecare drum succesorul unui număr se află pe rândul de mai jos și anume dedesubt sau pe diagonală la dreapta;

numărul de rânduri este strict pozitiv, dar cel mult egal cu 100;

toate numerele ce apar în triunghi sunt întregi cuprinşi între 0 şi 99.

Se vor afişa şi coordonatele numerelor ce intră în componenţa sumei obţinute.

Exemplu. Pentru triunghiul

```
   7
  3 8
 8 1 0
   7   4
4 5 2 6
```

valoarea căutată este 30.

206. Într-o variantă a binecunoscutului joc de calculator "Lemmings" ecranul (de formă dreptunghiulară) este împărţit în mxn pătrăţele de dimensiuni egale, unele dintre ele fiind libere iar altele fiind ocupate cu zid. În punctul de coordonate (x1,y1) al ecranului este plasat un Lemming care are ca sarcină să ajungă cât mai repede în pătratul de coordonate (x2,y2). Se ştie că Lemming-ul se poate deplasa pe orizontală sau verticală cu o căsuţă, trecând într-o căsuţă liberă într-o unitate de timp, sau într-o căsuţă ocupată cu zid în 2 unităţi de timp (1 unitate de timp este necesară spargerii zidului). Afişaţi traseul pe care trebuie să-l parcurgă Lemming-ul.

Exemplu. Dacă ecranul este $\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{pmatrix}$ atunci traseul optim de la (1,1) la (4,7) este (1,1),(1,2),(1,3),(1,4),(1,5),(2,5),(3,5),(4,5),(4,6),(4,7) consumând 10 timpi, iar de la (3,3) la (1,7) este (3,3),(3,4),(3,5),(2,5),(2,6),(2,7),(1,7) adică sunt necesari 8 timpi.

207. Un şoricel se găseşte într-un labirint codificat sub forma unei matrice cu m linii şi n coloane, fiecare pătrăţel având una din valorile: 0 – pentru culoar; 1 - pentru zid, 2 – pentru brânză, 3 – pentru poziţia iniţială a şoricelului. Să se găsească un traseu de lungime minimă pe care trebuie să-l parcurgă şoricelul pentru a ieşi din labirint (a ieşi în afara matricei). Din toate traseele de lungime minimă se va afişa cel pentru care cantitatea de brânză adunată de şoricel este maximă. Şoricelul se poate deplasa doar orizontal sau vertical.

Datele de intrare se citesc de la tastatură. Afişarea traseului se va face pe ecran sub forma unui şir de caractere din mulţimea {N,S,E,V}, reprezentând

drumul parcurs de şoricel. Se va afişa apoi şi numărul de bucăţele de brânză mâncate de şoricel.

Exemplu. Pentru labirintul

$$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 2 & 2 & 1 & 2 & 1 & 0 & 1 & 2 & 2 \\ 1 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 0 \\ 0 & 1 & 2 & 0 & 3 & 0 & 1 & 1 & 0 \\ 2 & 0 & 1 & 0 & 2 & 0 & 0 & 0 & 1 \\ 1 & 0 & 2 & 2 & 1 & 2 & 1 & 2 & 0 \\ 0 & 1 & 2 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

traseul de lungime minimă este **NVNN**, iar cantitatea de brânză mâncată de şoricel este 3.

208. Între două maluri ale unei văi adânci s-a construit un pod suspendat format din N bucăţi de scândură, legate cu liane. Vom considera că scândurile sunt numerotate de la 1 la N, începând de pe malul pe care ne aflăm. În timp unele bucăţi de scândură s-au deteriorat, iar altele chiar au dispărut. Pentru traversarea podului se ştie că:

se pot face paşi doar de lungime 1, 2 sau 3;

- scândurile deteriorate sunt nesigure, deci pe ele şi de pe ele se pot face doar paşi de lungime 1.
- evident, nu se poate păşi pe o scândură care lipseşte.

Scrieţi un program care să determine numărul de modalităţi de traversare a podului (mai exact, de a ajunge pe celălalt mal), precum şi o soluţie de traversare, dacă o astfel de soluţie există. La afişarea unei soluţii se vor indica numerele de ordine ale scândurilor pe care se calcă.

Exemplu. Pentru N=10, 2 scândurile 2 şi 7 lipsă şi scândura 5 deteriorată există un număr de total de 48 de posibilităţi, o soluţie fiind (3,6,8). Pentru N=6, scândurile lipsă 2 şi 4, şi scândura 3 deteriorată, nu există nici o soluţie.

209. Un teren dreptunghiular, de dimensiune m×n, este împărţit în pătrate identice, având lungimea laturii 1 (0<m<101, 0<n<101). Fiecărui pătrat îi este asociat un punctaj (număr întreg între 0 şi 99). Un copil trebuie să ajungă de la latura de sus la latura de jos astfel încât să aibă un punctaj total minim (punctaj calculat prin însumarea punctajelor pătratelor prin care trece). Ştiind că din pătratul (i,j) are voie să treacă, dacă este posibil (fără să iasă din dreptunghi), în unul dintre punctele:

a) (i+1,j-1), (i+1,j), (i+1,j+1)

b) (i,j-1), (i+1,j-1), (i+1,j), (i+1,j+1)

c) (i+1,j-1), (i+1,j), (i+1,j+1), (i,j+1)

d) (i,j-1), (i+1,j-1), (i+1,j), (i+1,j+1), (i,j+1)

şi cunoscând punctul de pornire, se cere punctajul minim şi un traseu de punctaj minim.

Exemplu. Pentru `m=6, n=6` şi terenul $\begin{pmatrix} 5 & 0 & 2 & 8 & 3 & 3 \\ 5 & 1 & 1 & 1 & 0 & 4 \\ 7 & 8 & 10 & 6 & 1 & 3 \\ 2 & 9 & 6 & 3 & 1 & 9 \\ 4 & 1 & 1 & 1 & 4 & 5 \\ 5 & 0 & 7 & 2 & 9 & 3 \end{pmatrix}$ şi poziţia de start `(1,2)` se obţin următoarele soluţii:

a) `(1,2), (2,2), (3,1), (4,1), (5,2), (6,2)`, cost `11`.

b) `(1,2), (2,3), (3,4), (4,5), (5,4), (5,3)->(6,2)`, cost `10`.

c) `(1,2), (2,3), (2,4), (3,5), (4,4), (5,3), (6,2)`, cost `7`.

d) `(1,2), (2,3), (2,4), (3,5), (4,5), (5,4), (5,3), (6,2)`, cost `6`.

210. Se dă o matrice `A` de dimensiune `n×n` cu numere întregi strict pozitive. Se cere să se determine probabilitatea ca plecând din punctul `(1,1)` să se ajungă în punctul `(n,n)`, ştiind că din punctul `(i,j)` se pot ajunge doar în punctul `(i,j+1)` cu probabilitatea $1/A[i,j]$ sau în punctul `(i+1,j)` cu probabilitatea $1 - 1/A[i,j]$.

Exemplu. Pentru matricea $\begin{pmatrix} 2 & 3 & 2 & 1 \\ 5 & 3 & 4 & 2 \\ 1 & 2 & 1 & 2 \\ 1 & 3 & 2 & 1 \end{pmatrix}$ probabilitatea va fi `0.329282`.

211. Se considera rombul

$$\begin{array}{ccccccc} & & & a_1 & & & \\ & & a_2 & & a_3 & & \\ & a_4 & & a_5 & & a_6 & \\ a_{n(n-1)/2+1} & & a_{n(n-1)/2+2} & & & & a_{n(n+1)/2} \\ & & & a_{n*n-2} & & a_{n*n-1} & \\ & & & & a_{n*n} & & \end{array}$$

unde a_i cu `i=1,n*n` sunt numere naturale citite din fisierul `in.txt`. Cu ajutorul acestui romb se pot forma sume in felul următor:

se porneşte cu numărul de pe linia `1`

succesorul unui număr se afla pe linia următoare plasat în diagonala

Care este cea mai mare sumă care se poate forma astfel şi care sunt numerele care o alcătuiesc? Se cere determinarea unei singure variante.

Exemplu. Pentru şirul `4, 1, 2, 5, 7, 3, 1, 14, 12, 4, 2, 6, 9, 3, 2, 11, 7` suma maximă este `49` obţinută astfel

```
        1
      2
    7   3   1
14  12    4   2
    6   9   3
      2  11
        7
```

212. O zonă mlăştinoasă are formă dreptunghiulară, având `nl` linii şi `nc` coloane. Ea este formată din celule cu latura de o unitate. O parte din acestea reprezintă uscat, iar altele reprezintă apă, uscatul fiind codificat cu `0`, iar apa cu `1`. Se doreşte să se obţină un drum de pe malul de nord spre cel de sud, trecând doar pe uscat. Celulele cu apă pot fi transformate în uscat, paraşutând într-un loc cu apă câte un ponton (o plută) de dimensiunea unei celule. Deoarece paraşutarea este periculoasă, se doreşte să avem un număr minim de paraşutări. Deplasarea se poate face cu câte o celulă pe linie, pe coloană, sau pe diagonală.

Scrieţi un program care determină numărul minim de pontoane şi coordonatele acestora.

Fişierul de intrare `LAC.IN` are următoarea structură :

- pe prima linie se află două numere naturale `nl` şi `nc` (`1<=nl,nc<=100`) separate de un spaţiu, reprezentând numărul de linii, respectiv de coloane ale zonei;
- pe următoarele `nl` linii se află câte `nc` valori binare separate de câte un spaţiu, reprezentând configuraţia zonei (`0` pentru uscat şi `1` pentru apă)

Fişierul de ieşire LAC.OUT va avea următoarea structură:

- pe prima linie un număr natural `min`, care reprezintă numărul cerut de pontoane;
- pe următoarele `min` linii vom avea câte două numere naturale separate de câte un spaţiu, reprezentând coordonatele celor `min` pontoane (linie şi coloană).

Dacă există mai multe soluţii se va afişa una singură.

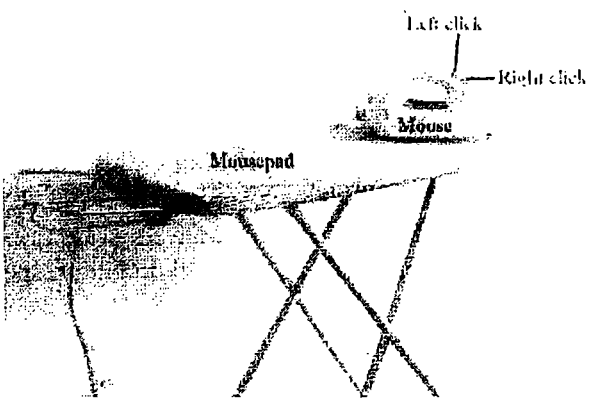
Exemplu.

LAC.IN

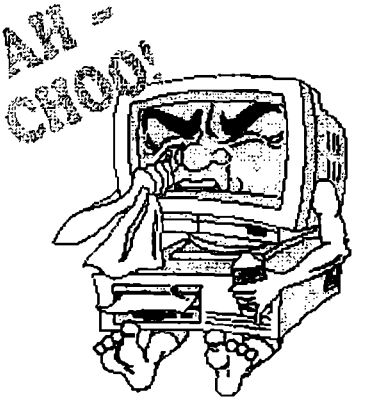
```
8 9
0 1 1 1 1 1 1    1
0 1 1 1    1 1    1
1 0 1 1 1 0 1 1 1
1 1 0 0 1 1 0 1 1
1 1 1 1 1 1 1 0 1
1    1 1    1    1 0
                1
        1 1          1
```

LAC.OUT

```
2
4 5
  8
```



8. Structuri dinamice de date



8.1. Liste

8.1.1. Liste simplu înlănțuite

1. Știind că procedura **new** returnează **NIL** dacă nu mai există spațiu în **HEAP**, scrieți un program care determină dimensiunea spațiului **Heap** liber.
2. Fiind dată o listă liniară simplu înlănțuită să se scrie un algoritm care va număra nodurile din listă.
3. Să se creeze o listă liniară simplu înlănțuită, fiecare nod conținând câte un număr întreg, și apoi:
   a) să se afișeze minimul dintre cheile din listă;
   b) afișează cheile de pe pozițiile impare (prima, a treia, a cincea, etc);
   c) afișează cheia nodului cu numărul de ordine `k`;
   d) verifică dacă cheile listei sunt în ordine descrescătoare;
   e) afișează numărul de apariții a unei valori date `k`.
4. Scrieți un algoritm care va schimba câmpul `INFO` din al `k`-lea nod cu valoarea dată `y`, într-o listă liniară simplu înlănțuită.
5. Fișierul `NUMERE.TXT` conține pe fiecare linie câte un număr întreg. Să se creeze și apoi să se tipărească o listă liniară simplu înlănțuită în care se vor introduce numai valorile pare din fișierul dat.

   Exemplu. Dacă fișierul conține numerele `2, 3, 6, 5, 10, 1`, câte unul pe o linie atunci lista va conține numerele `2, 6, 10`.
6. Se citește de la tastatură un număr întreg cu maxim opt cifre. Să se afișeze oglinditul numărului folosind o listă liniară simplu înlănțuită în nodurile căreia se memorează cifrele numărului.
7. Se citește un șir de cuvinte terminat cu '`*`' și se cere să se afișeze aceste cuvinte în ordine inversă celei de la citire, fără a utiliza șiruri sau subprograme recursive.

8. Se dau două liste liniare simplu înlănțuite, informația din fiecare nod reprezentând un număr real. Să se construiască alte două liste liniare simplu înlănțuite care vor conține elementele comune celor două liste, respectiv elementele care se găsesc în prima listă și nu se găsesc în a doua listă.

   Exemplu. Dacă prima listă conține numerele `2,5,3,10,22,4`, iar a doua `1,5,8,4,10` atunci în final prima listă va conține numerele `5,10,4`, iar a doua `2,3,22`.

9. Se citesc din fișierul `IN.TXT` datele despre notele la teza de informatică a unei clase de elevi. Fișierul conține:

   pe prima linie numărul `n` de elevi;

   pe fiecare din următoarele `n` linii, media, apoi numele unui elev, separate prin spațiu.

   Folosind o listă liniară simplu înlănțuită, se cere:

   a) să se afișeze numele elevului (elevilor) care au obținut nota cea mai mare;

   b) să se determine media generală a clasei.

10. Să se scrie un program care verifică dacă două liste liniare simplu înlănțuite sunt identice sau nu.

11. Se dă o mulțime de puncte în plan. Se cere:

   - să se creeze o listă simplu înlănțuită care conține coordonatele `(x,y)` ale punctelor date;
   - să se afișeze punctele din listă care se află în interiorul cercului $C(x_0, y_0, R)$, unde x_0, y_0, R se citesc de la tastatură.

   Exemplu. Dacă nodurile date sunt `(2,1)`, `(6,4)`, `(1,5)`, `(4,3)`, `(3,3)`, `(2,2)` iar $x_0=3$, $y_0=3$, $R=2$ atunci punctele care se vor afișa sunt: `(2,1)`, `(4,3)`, `(3,3)`, `(2,2)`.

12. Se dă o listă simplu înlănțuită care conține coordoatele `(x,y)` ale unor puncte din plan. Se cere:

   - Să se listeze punctele care se află pe prima bisectoare;
   - Să se elimine punctele aflate pe axa `Ox` sau `Oy` și apoi să se afișeze punctele rămase în listă;
   - Să se insereze înaintea punctelor care nu se află pe prima bisectoare, punctul simetric față de aceasta;

   Exemplu. Pentru lista `(19,3)`, `(3,3)`, `(-5,0)`, `(4,9)`, `(10,10)`, `(0,10)`

   a) se listează simplu `(3,3)`, `(10,10)`

   b) se elimină punctele `(-5,0)`, `(0,10)` deci lista rămasă va fi `(19,3)`, `(3,3)`, `(4,9)`, `(10,10)`.

   c) lista devine `(3,19)`, `(19,3)`, `(3,3)`, `(9,4)`, `(4,9)`, `(10,10)`.

13. Să se creeze o listă liniară simplu înlănțuită cu articole având structura:

```
Nume_student    An    Grupa    Media
```

Să se listeze toți studenții cu `Media>9`, apoi să se elimine din listă toți studenții din anul `5`.

14. Se dă o listă liniară simplu înlănțuită. Să se scrie o procedură care elimină elementele de pe pozițiile numere prime.

Exemplu. Dacă lista conține numerele `2,9,13,5,67,11,19,18` atunci în final lista va fi `2,5,11,18`.

15. Un centru de închiriat casete video întocmește o listă (simplu înlănțuită) a casetelor disponibile pentru închiriere. Pentru fiecare casetă vor fi trecute în listă titlul filmului, numele regizorului și durata în minute. Scrieți un program care în funcție de dorința utilizatorului realizează următoarele acțiuni:

adăugarea datelor aferente unei noi casete în listă;

ștergerea din listă a unei casete cu titlul dat de la tastatură;

afișarea întregii liste;

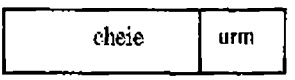
afișarea casetelor cu filmele regizate de un regizor cu nume introdus de la tastatură;

căutarea în listă a unui anumit film cu titlu dat de la tastatură.

16. Se dă o listă liniară simplu înlănțuită. Să se scrie o procedură care adaugă după fiecare element de pe pozițiile impare `p`, un număr de `p` elemente având ca și conținut valoarea elementului de pe poziția `p`.

Exemplu. Dacă lista inițială este `2,4,6,11`, după adăugare ea va conține `2,2,4,6,6,6,6,11`.

17. Fie dată o listă liniară având nodurile de forma:



unde cheie este un număr întreg, spre a cărei prim nod pointează variabila pointer `FIRST`. Lista este ordonată crescător după câmpul cheie astfel încât primul și ultimul nod conțin cea mai mică, respectiv cea mai mare valoare. Se cere să se scrie un program care șterge din listă un număr de noduri consecutive pentru care valoarea cheie este mai mare sau egală decât `KMIN` și mai mică sau egală decât `KMAX`.

Exemplu. Pentru `KMIN=25`, `KMAX=40` și lista `10->15->25->29-> 37->40->50`, va rezulta lista `10->15->50`.

18. Se dă șirul `x` cu `n` numere naturale citite de la tastatură. Să se creeze o listă liniară simplu înlănțuită conținând aceste numere. Să se elimine din listă elementele x_i pentru care suma elementelor $x_1, \ldots, x_{i-1}$ este egală cu x_i. Să se elimine apoi din listă elementele negative.

19. Să se scrie un subprogram care elimină unul sau două (dacă lista conține un număr par de noduri) elemente din mijlocul unei liste liniare simplu înlănțuite. Subprogramul primește ca parametru adresa primului element al listei.

Exemplu. Dacă lista inițială este `2->5->19->12->33` atunci în final ea va conține `2->5->12->33`.

20. Pentru organizarea unei selecții pentru un rolul principal feminin al unui film este necesară alcătuirea unei liste dinamice simplu înlănțuită, în care fiecare nod conține patru câmpuri de informație:

    numele și prenumele actriței

    culoarea părului

    culoarea ochilor

    nota acordată de comisia de selecție (formată exclusiv din bărbați ☺)

    a) Să se creeze lista cu toate actrițele care participă la selecție

    b) Să se listeze primele `n` concurente în ordinea descrescătoare a notelor.

    c) Din lista de la punctul a) să se șteargă fizic toate concurentele blonde cu ochii verzi și să se afișeze apoi concurentele rămase.

21. Presupuneți că aveți o listă liniară simplu înlănțuită spre începutul căreia pointează variabila pointer `START`. Fiecare nod conține o cheie de căutare `cheie`, un câmp de informații `info`, și un pointer `urm` spre următorul nod. Să se scrie o funcție `CAUT(x,DATA)` care simultan caută și reorganizează lista în felul următor. Un nod cu cheia de căutare `x` este căutat. Nodurile listei sunt examinate secvențial. Dacă nodul este găsit, el este șters din poziția sa curentă și este mutat la începutul listei. Informațiile conținute în câmpul `info` al nodului sunt returnate în parametrul `DATA`. Valoarea returnată de funcție este `TRUE` dacă elementul cerut a fost găsit și `FALSE` în caz contrar.

22. Se dă o mulțime de articole de forma:

    `(Nr_legitimație, Nume_student, Nota)`.

    Se cere:

    - să se creeze o listă simplu înlănțuită care conține aceste articole:
    - să se ordoneze articolele descrescător după `Nota`, prin modificarea legăturilor dintre articole în cazul în care este nevoie de intershimbare:
    - să se listeze articolele din lista sortată.

23. Se consideră un fişier text cu numele dat de la tastatură, ce conţine elementele unui şir $x_1, x_2, \ldots, x_n$ ($n \geq 1$, x_i de tip byte). Se cere:
    1) să se memoreze şirul într-o listă liniară simplu înlănţuită;
    2) să se listeze şirul memorat;
    3) să se determine un subşir de lungime maximă, cu elementele ordonate crescător (ordinea apariţiilor în şirul iniţial se păstrează şi în subşir);
    4) să se determine toate subşirurile de lungime maximă cu proprietatea de la punctul anterior.

    Exemplu. Pentru şirul iniţial: `1,7,2,3,9,4,5,12`, subşirul crescător de lungime maximă este `1,2,3,4,5,12`.

24. Se dă o listă simplu înlănţuită cu elemente de forma

```
TYPE  pnod=^nod;
      nod=record
            INFO:real;
            urm:pnod;
          end;
```

    unde `urm` este adresa elementului următor din listă. Să se scrie câte o procedură care tipăreşte elementele listei (`INFO`), de la cap la coadă şi invers fără a crea alte liste, fără a modifica lista dată şi printr-o singură parcurgere de fiecare dată.

25. Scrieţi o procedură recursivă care primeşte o valoare de tip caracter şi un pointer spre primul element al unei liste, având tipul unui nod definit mai jos. Procedura va afişa codurile corespunzătoare fiecărei apariţii a caracterului dat în ordinea în care apare în listă.

```
TYPE pnod=^nod;
     nod=record
           c:char;
           cod:integer;
           urm:pnod;
         end;
```

    Exemplu. Pentru lista `('c',1)->('o',56)->('c',23)->('a',99)` şi caracterul `'c'` se vor afişa valorile `1, 23`.

26. Scrieţi o procedură recursivă care primeşte o valoare de tip caracter şi un pointer spre primul element al unei liste având acelaşi tip ca şi în problema precedentă. Procedura va afişa codurile corespunzătoare fiecărei apariţii a caracterului dat în ordinea inversă apariţiei lui în listă.

    Exemplu. Pentru lista `('c',1)->('o',56)->('c',23)->('a',99)` şi caracterul `'c'` se vor afişa valorile `23,1`.

27. Să se scrie o funcție care va inversa o listă simplu înlănțuită. Presupunem că fiecare nod conține în câmpul de informație un caracter.

Parametrul funcției va fi un pointer către primul nod al listei inițiale. Funcția va returna pointerul spre primul nod al listei inversate:

Exemplu Lista inițială `'A'->'B'->'C'` va deveni `'C'->'B'->'A'`

Observație Nu se vor folosi procedurile `NEW` sau `DISPOSE`.

28. Scrieți o procedură recursivă care inserează o valoare într-o listă simplu înlănțuită ordonată. Parametrii procedurii sunt: un pointer spre primul nod al listei simplu înlănțuite și valoarea de înserat.

Observație Se va utiliza o astfel de recursivitate care să nu necesite instrucțiuni repetitive în corpul procedurii.

29. Se dau trei matrice rare (matrice cu un număr foarte mare de zero-uri). `A`, `B` și `C` de aceeași dimensiune `N×N`, fiecare dintre ele reprezentată cu ajutorul câte unei liste liniare simplu înlănțuite. O lista va conține în nodurile sale valorile diferite de zero din matrice, împreună cu linia și coloana corespunzătoare. Să se construiască lista care va reprezenta matricea `D`, unde `D=A+B*C`.

Exemplu. Pentru matricele $A = \begin{pmatrix} 1 & 0 & 2 \\ 2 & 0 & 0 \\ 0 & 5 & 0 \end{pmatrix}$, $B = \begin{pmatrix} 1 & 6 & 0 \\ 0 & 5 & 1 \\ 0 & 0 & 10 \end{pmatrix}$, $C = \begin{pmatrix} 0 & 8 & 0 \\ 0 & 0 & 0 \\ 15 & 0 & 1 \end{pmatrix}$ cele trei liste vor fi: `A: (1,1,1)->(2,1,3)->(2,2,1)->(5,3,2)`; `B: (1,1,1)->(6,1,2)->(5,2,2)->(1,2,3)->(10,3,3)`; `C: (8,1,2)->(10,3,1)->(1,3,3)`, iar lista finală va fi: `(1,1,1)->(8,1,2)->(2,1,3)->(17,2,1)->(1,2,3)->(150,3,1)->(5,3,2)->(10,3,3)` corespunzând matricei $\begin{pmatrix} 1 & 8 & 2 \\ 17 & 0 & 1 \\ 150 & 5 & 10 \end{pmatrix}$.

30. Se dau două mulțimi `A` și `B` a căror elemente, numere întregi, se memorează în două liste liniare simplu înlănțuite. Se cere să se scrie câte un program care calculează:

a) $C = A \cup B$ b) $C = A \cap B$

c) $C = A - B$ d) $C = (A - B) \cup (B - A)$

Se vor considera pentru fiecare problemă două situații:

1) listele sunt ordonate crescător;

2) listele nu sunt ordonate.

31. Se dau două polinoame `P(x)` şi `Q(x)` de grad `n`, respectiv `m`, reprezentate ambele sub forma unei liste liniare simplu înlănţuită, fiecare nod conţinând un coeficient al polinomului împreună cu puterea corespunzătoare a lui `x`. Să se calculeze:

- `P(x) + Q(x)`
- `P(x) * Q(x)`
- `P(Q(x))`

rezultatul memorându-se de asemenea într-o listă liniară simplu înlănţuită.

32. Considerăm următoarea reprezentare a termenilor unui polinom de 3 variabile `x, y, z`, folosind o listă simplu înlănţuită cu nodurile de forma:

puterea lui x	puterea lui y	puterea lui z	coef.	urm

Presupunem că lista este ordonată după următoarea regulă: termenul $D_1\ x^{A_1}y^{B_1}z^{C_1}$ va precede termenul $D_2\ x^{A_2}y^{B_2}z^{C_2}$ dacă $A_1 > A_2$ sau, dacă A-urile sunt egale şi $B_1 > B_2$, sau dacă A-urile şi B-urile sunt egale şi $C_1 > C_2$.

a) Să se implementeze o funcţie care inserează un nod într-o astfel de listă.

b) Scrieţi un algoritm care creează o listă liniară ordonată ca mai sus şi care nu conţine termeni duplicaţi.

c) Fiind date două polinoame a căror prim nod este referit de pointerii `P` şi respectiv `Q`, să se scrie o funcţie care determină suma celor două polinoame şi returnează pointerul spre lista reprezentând această sumă.

d) Fiind date două polinoame a căror prim nod este referit de pointerii `P` şi respectiv `Q`, să se scrie o funcţie care determină diferenţa celor două polinoame şi returnează pointerul spre lista reprezentând această diferenţă.

e) Fiind date două polinoame a căror prim nod este referit de pointerii `P` şi respectiv `Q`, să se scrie o funcţie care determină produsul celor două polinoame şi returnează pointerul spre lista reprezentând acest produs.

f) Fiind dat un polinom a cărui prim nod este referit de pointerul `P`, să se scrie un algoritm care parcurge lista şi calculează valoarea polinomului pentru punctul `(x,y,z)` citit de la tastatură.

g) Scrieţi un algoritm de împărţire a polinomului `P` la polinomul `Q`. Polinomul cât va fi plasat într-o listă liniară simplu înlănţuită a cărui prim nod este dat de variabila pointer `C`, în timp ce restul este memorat într-o listă înlănţuită a cărui prim nod are adresa memorată în `R`.

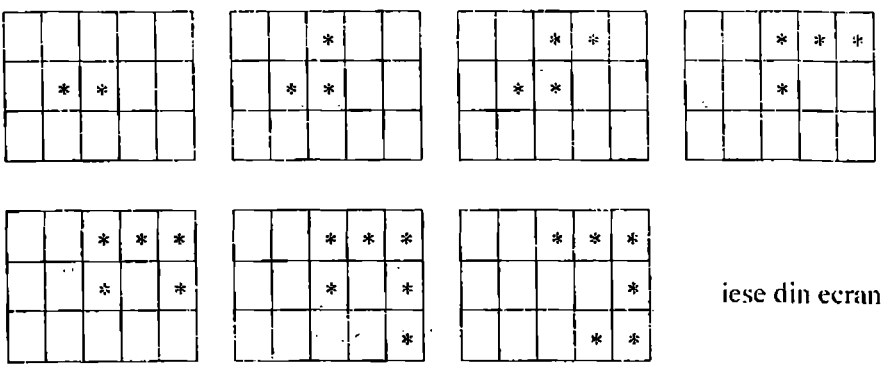
h) Scrieţi un algoritm care integrează un polinom de 3 variabile în funcţie de variabila `x`. Rezultatul va fi memorat într-o listă liniară simplu înlănţuită.

33. Se consideră un ecran de dimensiune `m*n` (`m≤25`, `n≤80`) şi o secvenţă de poziţii ale elementelor situate pe ecran având proprietatea că oricare două elemente succesive aflate pe aceeaşi linie sau coloană (fără spaţii între ele) sunt vecine. Elementele secvenţei sunt marcate cu `'*'` Se citesc poziţiile secvenţei, o valoare `k∈N` şi un şir de comenzi. Şirul poate să conţină comenzile `'N'`, `'S'`, `'E'`, `'V'`, al căror efect este adăugarea unui nou element `'*'` în poziţia corespunzătoare comenzii, faţă de ultimul element al secvenţei. Din `k` în `k` comenzi, primul element al secvenţei dispare. Se cere:

a) validarea datelor de intrare (a secvenţei de poziţii şi a şirului de comenzi);

b) să se semnaleze momentul când în urma unei comenzi se ajunge într-o poziţie existentă sau când în urma unei comenzi secvenţa ar ieşi din ecran.

Exemplu. Pentru `m=3`, `n=5`, `k=3`, secvenţa de poziţii `(2,2)`, `(2,3)`, şirul de comenzi `'NEESSVS'` se va semnala ieşirea din ecran:



iese din ecran

Observaţii. Pentru rezolvare nu se vor folosi vectori sau matrice, ci doar liste simplu înlănţuite.

34. Fie `n` un număr întreg, `n≥0`, citit de la tastatură. Să se scrie un program care afişează lista elementelor din X_n în ordine crescătoare, unde X_n este definită prin:

$$X_n = \{x = 2^i 3^j 5^k \mid i, j, k \geq 0 \text{ si } x \leq n\}$$

Indicaţie Notând cu $X = \bigcup_{n \geq 0} X_n$, aceasta admite următoarea definiţie:

1. $1 \in X$
2. dacă $x \in X$ atunci $2x,\ 3x,\ 5x \in X$
3. X conţine numai numerele ce satisfac axiomele 1 şi 2.

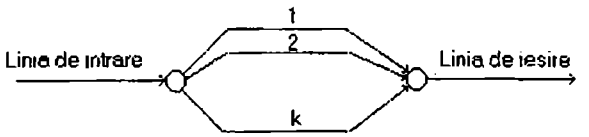
Deci se vor genera elementele $x \in X$ cu proprietatea că $x \leq n$.

35. Se consideră un depou de locomotive cu o singură intrare şi cu o singură linie ferată, care poate cuprinde oricâte locomotive. Să se scrie un program care realizează dispecerarea locomotivelor din depou. Programul prelucrează comenzi de forma: ***I*** (intrarea unei locomotive), ***E*** (ieşirea unei locomotive), ***L*** (listarea locomotivelor din depou) şi ***S*** (sfârşitul programului). La terminarea programului trebuie să se afişeze locomotivele existente în depou.

    Locomotivele pot intra şi ieşi din depou numai prin intrarea depoului, determinând în program o structură dinamică de tip stivă.

36. Aceeaşi problemă cu diferenţa că depoul are intrarea la un capăt şi ieşirea la capătul opus. Locomotivele intră în depou pe la intrare şi ies pe la ieşire, în aceeaşi ordine în care au intrat.

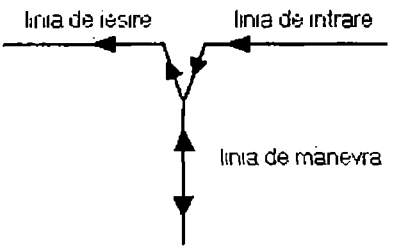
37. Pe o linie de cale ferată se găsesc, într-o ordine oarecare, `N` vagoane numerotate de la `1` la `N`. Linia se continuă cu alte `k` linii de manevre, ca în figura de mai jos. Cunoscând ordinea iniţială a vagoanelor, să se obţină la ieşire vagoanele în ordinea `1,2,...,N`; o linie este suficient de lungă astfel încât să încapă pe ea toate vagoanele.



38. Într-un triaj există o linie de cale ferată pentru manevre, ca în figura de mai jos, pe linia de intrare sunt `n` vagoane: `1,2,...,n`. Pe linia de intrare şi de ieşire deplasarea se poate face numai în sensul indicat de săgeţi; pe linia de manevră deplasarea se poate face în ambele sensuri.

    a) cunoscând ordinea vagoanelor la intrare şi ordinea în care se doreşte să apară ele la ieşire, să se afişeze, dacă problema are soluţie, manevrele necesare.

    b) aceeaşi problemă ca la punctul a), în condiţiile în care şi pe linia de intrare este permisă deplasarea în ambele sensuri;

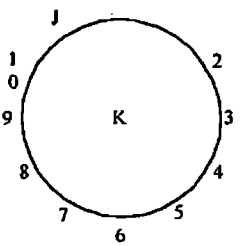


39. Pentru triajul din figura anterioară având la intrare vagoanele `1,2,...,n` într-o ordine oarecare să se afişeze toate posibilităţile lor de obţinere pe linia de ieşire.

40. Se citeşte un şir de litere mici ale alfabetului englez, terminat cu caracterul '$'. Se cere să se pună într-o stivă alocată dinamic toate literele citite. Se citeşte, apoi, o literă y din alfabet. Să se genereze două structuri dinamice de tip stivă, prima cuprinzând literele din stiva iniţială care preced în alfabet litera y, cealaltă cuprinzând literele din stiva iniţială care succed litera y.

41. Să se simuleze, folosind structuri dinamice de date, jocul "gâscă roşie" jucat de doi jucători. Un pachet de 32 de cărţi este împărţit în mod egal celor doi jucători. Cărţile se ţin cu faţa în jos, jucătorii pun alternativ pe masă cu faţa în sus cartea care se află deasupra pachetului lor. Dacă s-a pus o carte roşie, celălalt jucător trebuie să ia întregul pachet de cărţi de pe masă şi-l pune sub pachetul său. Jocul continuă în acest fel până când unul din jucători rămâne fără cărţi. Acesta câştigă jocul.

42. Să se implementeze cu ajutorul alocării dinamice a memoriei următorul joc de cărţi: Cărţile sunt împărţite în cerc, cu faţa în jos, formând un ceas, cu o stivă în dreptul fiecărei ore şi o stivă suplimentară în mijlocul ceasului. Astfel se formează 13 stive, cu 4 cărţi fiecare.



Apoi jocul începe. Cartea din vârful stivei "K" (ultima carte împărţită) este întoarsă cu faţa devenind cartea curentă. Fiecare din mutările următoare constă în plasarea cărţii curente, cu faţa în sus, sub stiva corespunzătoare valorii ei, şi întoarcerea, cu faţa în sus a cărţii aflate în vârful stivei respective. Aceasta va devenii cartea curentă. De exemplu dacă un "As" este cartea curentă, ea va fi pusă sub stiva de la ora 1, şi cartea din vârful stivei devine carte curentă. Jocul se termină când stiva indicată de cartea curentă nu mai conţine nici o carte cu faţa în jos.

Intrarea. Constă din patru linii conţinând fiecare câte 13 cărţi, separate între ele printr-un spaţiu. Fiecare carte este reprezentată prin două caractere, prima este valoarea cărţii (A,2,3,4,5,6,7,8,9,T,J,Q,K, unde T reprezintă valoarea 10) urmată de tipul cărţii (I – inimă roşie, R – romb, T – treflă, N – inimă neagră). Cărţile sunt listate de jos în sus, adică prima carte împărţită este ultima carte din fişierul de intrare.

Ieşirea. Constă dintr-o singură linie, conţinând numărul de cărţi cu faţa în sus existente la sfârşitul jocului, urmat de ultima carte care a fost întoarsă, în formatul dat la intrare.

Exemplu.

Intrarea	Ieşirea
TN QT 8N 8R QI 2R 3I KI 9I 2I TI KN KT 9R JI 7I JR 2N QN TR 2T 4I 5I AR 4R 5R 6R 4N 9N 5N 7N JN 8I 3R 8T 3N 4T 6N 9T AN 7T AI 6I KR JT 7R AT 5T TT QR 6T 3T	44 KR

43. Această problemă vă cere să simulaţi un alt joc de cărţi. Cărţile sunt puse pe masă una câte una într-o linie de la stânga la dreapta, fără a le suprapune. Când o carte se potriveşte cu cartea aflată în vârful stivei din stânga sa sau se cu cartea din a treia stivă din stânga sa, se va muta în vârful stivei respective. Spunem că două cărţi se potrivesc dacă ele au aceeaşi valoare sau sunt de acelaşi tip. Doar cartea din topul unei stive se poate muta. Spaţiul care se poate crea între stive se va elimina prin mutarea tuturor stivelor din dreapta spaţiului creat cu o poziţie spre stânga. Jocul continuă până când toate cărţile sunt puse pe masă.

    În cazul în care mai multe cărţi pot fi mutate, se va muta cea mai din stânga carte posibilă. Dacă o carte se poate muta atât cu o poziţie cât şi cu trei poziţii, ea se va muta trei poziţii.

    Intrarea. Intrarea constă din câte patru linii, fiecare linie conţinând 13 de cărţi separate printr-un spaţiu. Linia finală a fişierului de intrare conţine caracterul #.. Cărţile sunt este reprezentate prin câte două caractere, prima este valoarea cărţii (`A,2,3,4,5,6,7,8,9,T,J,Q,K`, unde `T` reprezintă valoarea `10`) urmată de tipul cărţii (`I` – inimă roşie, `R` – romb, `T` – treflă, `N` – inimă neagră).

    Ieşirea. Pentru fiecare pachet de cărţi (patru linii de intrare), se va afişa o linie, indicând numărul de cărţi rămase în fiecare stivă.

    Exemplu.

    Intrarea

```
QR AR 8I 5N 3I 5I TT 4R JI KN 6I 8N JN
AT AN 8R 2I QN TN 3N AI 4I TI TR 3T 6N
8T 7R 4T 4N 7N 9I 7T 5R 2N KR 2R QI JR
6R 9R JT 2T KI 3R QT 6T 9N KT 7I 9T 5T
AI 2I 3I 4I 5I 6I 7I 8I 9I KI 6N QI TI
AN 2N 3N 4N 5N JI 7N 8N 9N TN JN QN KN
#
```

    Ieşirea

```
6 stive ramase: 40 8 1 1 1 1
1 stiva ramasa: 52
```

44. Pe o tijă verticală sunt `n` bile colorate cu cel mult `k` culori, fiecare bilă având o etichetă cu un număr de la `1` la `n`. Să se mute bilele pe alte `k` tije, pe fiecare punând numai bile de aceeaşi culoare. Fiecare tijă are un capăt liber pe unde se introduc sau extrag bilele, celălalt capăt fiind sudat pe suportul tijei. Să se afişeze bilele de pe fiecare din cele `k` tije (se vor utiliza structuri dinamice).

45. Scrieţi un program care simulează funcţionarea unei cozi. În mod repetat, cât timp utilizatorul doreşte acest lucru, programul va permite alegerea uneia din operaţiile aplicabile cozii: adăugarea unui element în coadă, eliminarea unui element, afişarea cozii.

46. Într-un super-market sunt `n` case de marcat, numerotate `1,2,...,n`. Scrieți un program care simulează acțiunile a `m` cumpărători, numerotați `1,2,...,m`, știind că în magazin lucrurile se desfășoară după următorul algoritm:

    din `t` în `t` secunde, câte un cumpărător dorește să plătească produsele achiziționate în regim de auto-servire, și pentru aceasta se așează la coadă la casa unde sunt cei mai puțini cumpărători: ordinea în care cumpărătorii se așează la coadă este cea naturală, dată de numerotarea lor;

    pentru fiecare cumpărător se cunoaște timpul necesar plății produselor achiziționate, exprimate în secunde (care depinde evident de numărul produselor cumpărate);

    după ce a plătit, fiecare cumpărător părăsește coada de așteptare.

    Datele de intrare se citesc din fișierul '`MAGAZIN.IN`', care conține:

    pe primul rând valorile `m`, `n` și `t`, separate prin spații;

    pe fiecare din următoarele `m` rânduri câte un întreg, reprezentând timpul necesar unui cumpărător pentru achitarea produselor achiziționate.

    Programul va afișa "starea caselor de marcat" după fiecare intrare sau ieșire a unui cumpărător. Rezultatele se scriu în fișierul '`CASE.TXT`', în care o intrare/ieșire a unui cumpărător este înregistrată pe două rânduri astfel:

    pe primul rând se scriu trei valori: prima valoare va fi caracterul '`i`' sau '`e`' indicând dacă s-a produs o intrare sau o ieșire a unui cumpărător, a doua valoare reprezintă numărul casei unde s-a produs "evenimentul", iar a treia valoare desemnează numărul de ordine al cumpărătorului intrat/ieșit;

    pe al doilea rând se va reprezenta modul cum arată, după intrarea/ieșirea respectivă, coada de așteptare din fața casei de marcat la care a avut loc evenimentul (se vor scrie numerele de ordine ale cumpărătorilor aflați în coadă separate prin spații).

47. Pe o masă se găsesc `n` blocuri numerotate de la `1` la `n`, unul lângă altul. Un robot primește o serie de comenzi de forma:

    a) `A i j`

    unde `i` și `j` sunt numere de blocuri. La această comandă robotul returnează toate blocurile care sunt stivuite deasupra blocurilor `i` și `j` la pozițiile lor inițiale iar apoi mută blocul `i` peste blocul `j`.

    b) `B i j`

    returnează blocurile aflate deasupra lui `i` la pozițiile lor inițiale, iar apoi pune blocul `i` în vârful stivei care conține blocul `j`.

c) `C i j`

toate blocurile stivuite deasupra lui `j` sunt mutate la locurile lor inițiale apoi cele din stiva de deasupra blocului `i` se mută deasupra blocului `j` menținând ordinea

d) `D i j`

mută stiva de blocuri care începe cu `i` și toate blocurile de deasupra lui peste stiva de blocuri care conțin blocul `j`.

e) `Q`

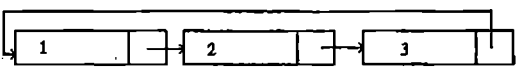
termină șirul de acțiuni.

Dându-se numărul inițial de blocuri și un șir de astfel de comenzi, să se afișeze modul de poziționare a blocurilor în final. Se vor folosi tipuri de date alocate dinamic.

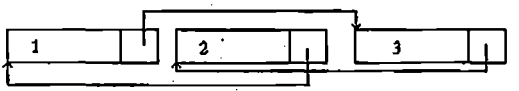
8.1.2. Liste circulare

48. Scrieți un algoritm de concatenare a două liste circulare simplu înlănțuite.
49. Elaborați un algoritm care să considere o listă circulară simplu înlănțuită și să inverseze direcțiile tuturor săgeților.

Exemplu. Din lista:



se va obține lista



50. Se dă o listă circulară simplu înlănțuită, ale cărei noduri conțin coordonatele unor puncte în plan. Să se modifice referințele la elemente astfel încât noua listă să reprezinte vârfurile în ordine trigonometrică, a unei linii poligonale închise care trece prin toate punctele date o singură dată.
51. Scrieți un algoritm de împărțire a unei liste circulare simplu înlănțuită în două liste circulare având un număr cât mai apropiat de elemente (dacă e posibil egal).

51. Scrieți un algoritm de împărțire a unei liste circulare simplu înlănțuită în două liste circulare având un număr cât mai apropiat de elemente (dacă e posibil egal).

52. Să se implementeze o variantă modificată a binecunoscutului joc "Nu te supăra frate" Tabla de joc este reprezentată de o listă circulară simplu înlănțuită, fiecare nod al listei conținând un număr natural pozitiv. Un singur nod conține valoarea zero. Acest nod este considerat ieșirea din joc. Ambii jucători pornesc din același nod. Un jucător aflat la mutare se deplasează cu `nr` căsuțe, unde `nr` este numărul memorat în nodul în care se află jucătorul. Valoarea din căsuța părăsită se modifică astfel încât noua valoare să fie egală cu partea întreagă a mediei aritmetice a valorilor memorate în cele două noduri vecine nodului respectiv. Jucătorii mută alternativ. Jocul se termină atunci când un jucător a ajuns la ieșire, acesta fiind declarat câștigător, sau dacă ambii jucători au efectuat `m` mutări fără a ajunge la ieșire. În acest caz este declarat câștigător jucătorul aflat cel mai aproape (în sensul legăturilor dintre nodurile listei) față de ieșire.

8.1.3. Liste liniare dublu înlănțuite

53. Se citește un șir de cuvinte terminat cu '*'. Să se afișeze cuvintele în ordine inversă citirii și, apoi, în ordinea în care au fost citite, fără a utiliza șiruri.

54. Să se scrie un program care tratează comenzile de generare și prelucrare a unei liste dublu înlănțuite cuprinzând cuvinte. Comenzile afișate de o procedură meniu sunt:

    inserarea în fața unui cuvânt dat;
    inserarea după un cuvânt dat;
    ștergerea unui cuvânt dat;
    traversarea listei de la început;
    traversarea listei de la sfârșit;
    oprirea programului.

    În cazul în care cuvântul după care, sau înaintea căruia trebuie să se facă introducerea nu este găsit, elementul nou se adaugă la sfârșitul listei.

55. Se dă o listă liniară dublu înlănțuită, informația din fiecare nod reprezentând o valoare numerică reală. Să se modifice referințele la elementele listei astfel încât noua înlănțuire să reprezinte ordonarea descrescătoare a elementelor listei.

56. Să se scrie un subprogram care determină numărul de elemente dintr-o listă liniară dublu înlănțuită. Subprogramul va primi ca parametru adresa unui element oarecare al listei.

57. Scrieți o funcție care construiește o listă liniară dublu înlănțuită dintr-o listă simplu înlănțuită care va fi accesată printr-un pointer `FIRST`.

58. Se dă o listă liniară dublu înlănțuită, având ca informație, în fiecare nod, o valoare numerică întreagă. Să se descompună această listă în trei liste astfel:

    prima să conțină elementele listei inițiale care dau `0` la împărțirea la `3`;

    a doua să conțină elementele care dau restul `1` la împărțirea cu `3`;

    și a treia listă conține restul elementelor din lista inițială.

    Exemplu. Dacă lista inițială conține numerele `12,9,5,7,91,3,17,22` atunci cele trei liste vor conține `12,9,3`; a doua `7,91,22`; respectiv `5,17` ultima.

59. Se dă o listă liniară dublu înlățuită, informația din noduri fiind numere întregi. Să se modifice referințele la nodurile listei astfel încât să se obțină două liste liniare dublu înlănțuite care vor conține elementele de pe pozițiile impare, respectiv pare din lista inițială.

    Exemplu. Dacă lista inițială conține, pe rând, numerele `3,7,12,9,10, 21,14` atunci prima listă rezultată va conține `3,12,10,14`, iar a doua `7,9,21`.

60. Să se scrie un program care creează o listă liniară dublu înlănțuită cu numere reale. Să se insereze apoi între oricare două noduri ale listei un nod nou conținând ca informație media aritmetică din cele două noduri.

    Exemplu. Dacă în lista inițială sunt memorate numerele `2,9,12,6,1` atunci în final lista va conține `2,5.5,9,10.5,12,9,6,3.5,1`.

61. Să se scrie un program care realizează următoarele cerințe. Opțiunile se vor include într-un meniu din care utilizatorul va avea posibilitatea să aleagă operația dorită.

    a) Să se creeze o listă liniară dublu înlănțuită în care, fiecare nod, pe lângă informațiile de legătură, va conține date referitoare la un elev candidat la examenul de capacitate: `nume`, `prenume` și un vector cu 3 componente care reține notele elevului la cele trei probe ale examenului (numere reale). Datele se vor citi de la tastatură.

    b) Să se afișeze pe ecran `numele`, `prenumele` și `media` la capacitate a fiecărui elev din lista creată.

    c) Să se calculeze media generală a tuturor candidaților care se găsesc în listă.

    d) Să se memoreze informațiile din listă în fișierul text "`CAPAC.TXT`" Prima linie a fișierului va conține numărul candidaților. Fiecare din următoarele linii va conține informațiile referitoare la un elev.

e) Să se citească datele din fişierul text creat anterior şi recreează lista.

f) Să se adauge înregistrări noi la sfârşitul listei. Datele se citesc dintr-un fişier text care are aceeaşi structură ca la punctele anterioare şi numele citit de la tastatură.

g) Să se inverseze două noduri consecutive din listă, noduri având numărul de ordine dat.

h) Folosind funcţia sau procedura scrisă la punctul anterior să se sorteze descrescător elementele listei după media fiecărui elev.

i) Să se listeze toţi elevii care au media mai mare sau egală cu o valoare citită de la tastatură.

j) Listaţi primii k elevi (în ordinea descrescătoare a mediilor). În cazul în care există mai mulţi elevi cu aceeaşi medie pe poziţia k se vor afişa toţi.

k) Să se elimine din listă elevii cu numele şi prenumele date de la tastatură.

62. Pentru a putea lucra cu numere întregi mari, care nu pot fi memorate în nici o variabilă de tip predefinit, se pot folosi listele liniare dublu înlănţuite, fiecare nod al unei liste memorând o cifră a acestui număr. Folosind astfel de liste se cere să se calculeze:

a) suma a două numere întregi mari;

b) scăderea a două numere întregi mari;

c) înmulţirea a două numere întregi mari;

d) împărţirea a două numere întregi mari;

e) compararea a două numere întregi mari;

f) maximul a n numere întregi mari;

g) factorialul unui număr întreg mare.

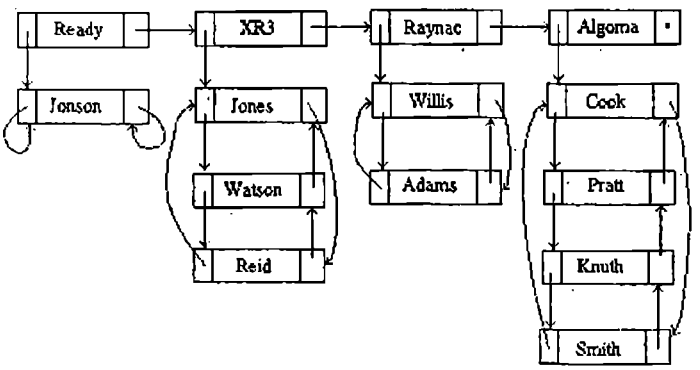
8.1.4. Structuri diverse de tip listă

63. Un grup este format din N+1 indivizi: pentru a transmite diverse mesaje, şeful grupului A_0 foloseşte următorul procedeu de comunicare: A_0 comunică cu A_1 şi A_2, A_1 cu A_2 şi A_3, ..., A_{n-1} şi A_n transmit mesajul de confirmare către A_0. Dacă un individ părăseşte grupul, legăturile se refac; dacă în grup apar indivizi noi, aceştia sunt plasaţi la sfârşitul lanţului de comunicare. Să se scrie un program care să simuleze acest sistem utilizând alocarea dinamică a datelor.

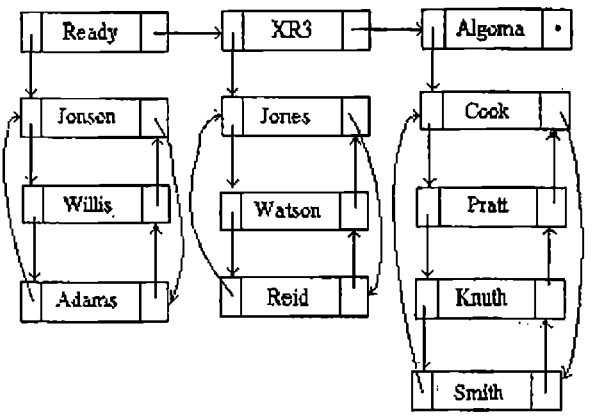
64. Pentru a ajuta la administrarea implementării unor proiecte software, un sistem de administrare a fost dezvoltat de firma **DataPro**. Prima sarcină a sistemului este să urmărească analiștii și programatorii la ce proiecte lucrează și care sunt disponibili pentru a lucra dar nu au fost încă asignați nici unui proiect.

Considerăm de exemplu structura de date din figura de mai jos. În figură, trei proiecte sunt curent active (**XR3**, **Raynac** și **Algoma**) și personalul pentru fiecare este listat dedesubtul numelui de cod al proiectului. Lista personalului disponibil pentru lucru este păstrat sub nodul proiect etichetat cu **Ready**.

Presupunem o structură care conține câmpurile **LPTR**, **Name** și **RPTR** unde **Ready** este nodul pointat de **First**.

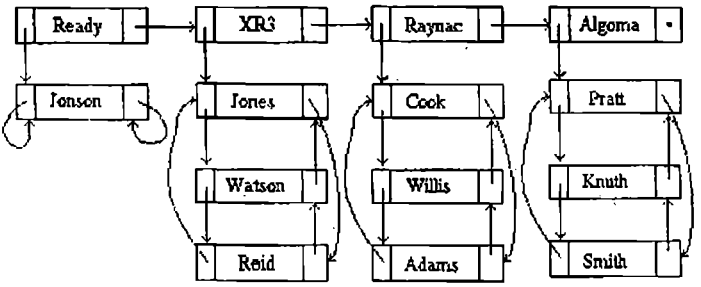


a) Scrieți un algoritm care rezolvă cazul în care un proiect este terminat - toate persoanele asociate proiectului sunt atașate listei **Ready**. De exemplu structura de mai sus se modifică obținând figura următoare atunci când proiectul **Raynac** este terminat:

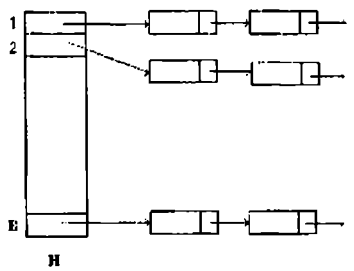


b) Scrieți un algoritm care mută un programator de la un proiect la alt proiect. De exemplu, dacă **Cook** este mutat de la proiectul **Algoma** la

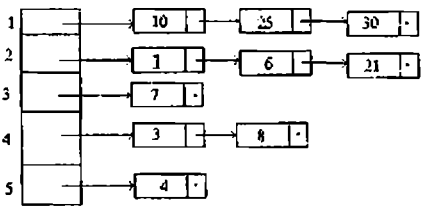
proiectul **Raynac**, pentru struetura inițială, atunci noua structură va arăta astfel ca în figura:



65. Pentru a memora un dicționar vom proceda după cum urmează: mulțimea membrilor potențiali ai dicționarului se împart într-un număr finit de clase. Dacă dorim să avem `B` clase, numerotate `1,2,...,B`, atunci vom folosi o funcție `h`, numită funcție de **hash**, astfel încât pentru un obiect `x` având tipul membrilor mulțimii, `h(x)` este unul din întregii `1,2,...,B`. Valoarea `h(x)` este clasa din care `x` va face parte. Fiecare clasă este organizată ca o listă liniară simplu înlănțuită. Pointerii spre primul element al fiecărei liste se vor păstra într-un tablou `H`.



(a)



(b)

a) Să se scrie o procedură care creează o astfel de structură, numită **open hashing**, în care se memorează numere întregi citite de la tastatură, știind că vom avea `5` clase iar funcția de hash va fi `h(x)=x mod 5+1`. În interiorul unei clase numerele vor fi păstrate în ordine crescătoare. De exemplu pentru șirul de intrare `1,7,10,21,4,30,8,25,6,3` structura creată va arăta ca în figura (`b`).

b) Fiind date două mulțimi `A` și `B` de numere întregi reprezentate folosind algoritmul de la punctul anterior să se scrie câte o procedură pentru fiecare din următoarele operații:

1) $A \cup B$

2) $A \cap B$

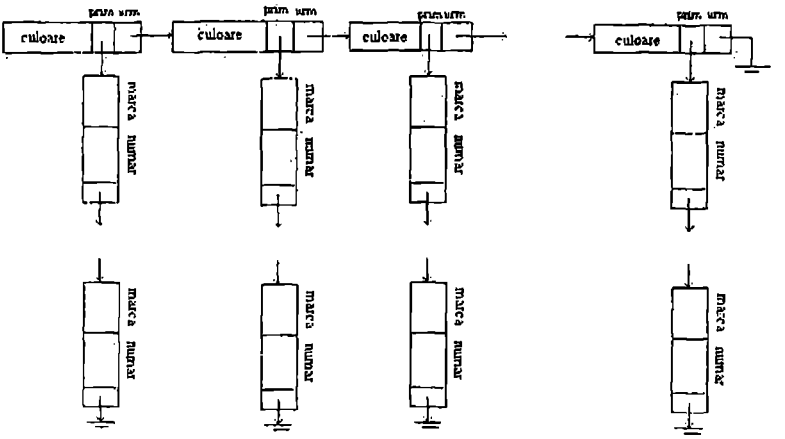
3) $A - B$.

c) Să se scrie o procedură care pornind de la o structură open hashing de tipul celei de la punctul a), să creeze o listă liniară simplu înlănțuită care să conțină toate numerele din structură în ordine crescătoare.

Observație. Nu se vor folosi procedurile `new` și `dispose` ci se vor realiza doar modificări ale adreselor conținute în nodurile structurii. Primul nod al listei finale va fi indicat de pointerul `H[1]`.

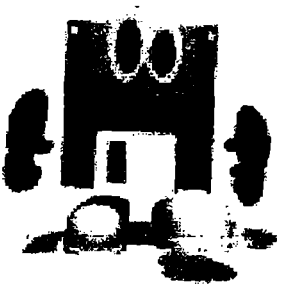
66. Se dă o mulțime de mașini caracterizate prin `culoare`, `marcă`, `număr`. Să se construiască o listă cu culorile mașinilor date și, pentru fiecare culoare, o sublistă a mașinilor având acea culoare. Să se listeze numerele de înmatriculare ale mașinilor care au o culoare și o marcă dată.

Indicație. Se va folosi o structură recursivă ca cea din figura de mai jos.



8.2. Arbori

8.2.1. Arbori binari



67. Să se creeze un arbore binar având drept chei numere întregi citite de la tastatură, apoi să se tipărească suma cheilor pare din arbore.

68. Să se scrie un program care calculează numărul de nivele (înălțimea) unui arbore binar.

69. Să se scrie un program care verifică dacă un arbore binar este arbore de căutare sau nu.

70. Scrieți un program care afișează cheia minimă și cheia maximă dintr-un arbore binar de căutare.

71. Să se calculeze produsul cheilor pozitive ale unui arbore binar.

72. Să se afişeze cheile din nodurile care au exact un succesor (fie stâng, fie drept).

73. Nodurile unui arbore binar conţin numere naturale. Să se scrie câte o funcţie recursivă pentru a determina:
    a) numărul total de noduri
    b) numărul de noduri terminale
    c) numărul de noduri care au "valoare" un număr care citit de la dreapta la stânga este acelaşi. (de exemplu 3443, 22, 3, 515)

74. Scrieţi o procedură care primeşte ca parametru pointerul spre rădăcina unui arbore binar şi afişează toţi descendenţii stângi ai rădăcinii (exclusiv rădăcina).

75. Scrieţi o procedură care primeşte ca parametrii pointerul spre rădăcina unui arbore binar cu cheile din noduri numere întregi şi doi întregi `x` şi `y`. Procedura va înlocui toate aparitiile valorii `x` în arbore, cu valoarea `y`.

76. Scrieţi o funcţie recursivă care primeşte o valoare întreagă şi un pointer spre rădăcina unui arbore binar cu nodurile conţinând numere întregi, şi returnează numărul componentelor egale cu întregul primit ca parametru.

77. Să se scrie un algoritm care determină dacă doi arbori `A` şi `B` sunt identici folosind una din cele 3 metode de traversare a arborilor.

78. Scrieţi un program care afişează cheile nodurilor aflate pe un anumit nivel dat.

79. Să se afişeze numărul cheilor pozitive şi numărul cheilor negative dintr-un arbore binar.

80. Să se scrie un program care construieşte un arbore binar cu cheile din noduri numere întregi. Numerele din oricare două noduri vor fi diferite. Dându-se apoi două numere `v` şi `w` programul verifică dacă acestea se află în arbore şi dacă `w` este descendent al lui `v`, afişând mesaje corespunzătoare.

81. Scrieţi un program care găseşte cel de-al `k`-lea element din traversarea în preordine a unui arbore binar.

82. Scrieţi un program care găseşte cel de-al `k`-lea element din traversarea în inordine a unui arbore binar.

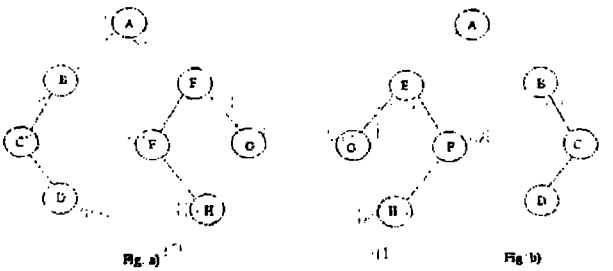
83. Scrieţi un program care găseşte cel de-al `k`-lea element din traversarea în postordine a unui arbore binar.

84. Se dau vectorii `x` şi `y` cu câte `n` componente numere întregi. Interpretând aceşti vectori ca fiind informaţiile din nodurile unui arbore binar parcurs în preordine (vectorul `x`) respectiv inordine (vectorul `y`) să se creeze arborele iar apoi să se afişeze pe ecran informaţiile din nodurile arborelui parcurs în postordine.

85. Dându-se vectorul `x` cu `n` componente numere întregi reprezentând informațiile din nodurile unui arbore binar de căutare parcurs în preordine, să se creeze arborele și să se parcurgă apoi în postordine și în inordine.

86. Scrieți un algoritm pentru a obține versiunea în oglindă a unui arbore binar.

Exemplu. Figura b) arată versiunea în oglindă a arborelui din figura a).



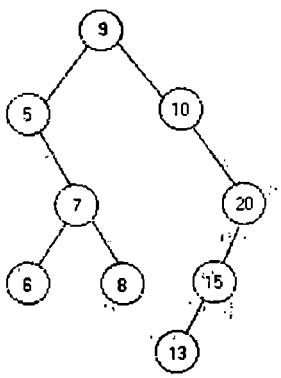
87. Să se scrie un algoritm care verifică dacă un arbore binar este simetric, adică el este identic cu imaginea sa în oglindă.

88. Se dă un arbore binar ale cărui chei sunt numere întregi, nu neapărat distincte. Să se creeze o listă liniară simplu înlănțuită având în noduri cheile distincte ale arborelui.

89. Se dă un arbore binar ale cărui chei sunt numere întregi, nu neapărat distincte. Să se creeze o listă liniară simplu înlănțuită cu cheile arborelui în ordinea crescătoare (fiecare cheie a arborelui se va memora într-un nod în listă, iar lista se va crea de la început ordonată, fără a aplica ulterior vreun algoritm de ordonare).

90. Dându-se un arbore binar de căutare se cere să se scrie un algoritm de liniarizare a acestuia, adică să se transforme arborele într-o listă liniară dublu înlănțuită ordonată.

Exemplu. Arborele din figura de mai jos se va transforma în lista `5 ↔ 6 ↔ 7 ↔ 8 ↔ 9 ↔ 10 ↔ 13 ↔ 15 ↔ 20`.



91. Pentru evidența cărților dintr-o bibliotecă se definește un arbore binar. Fiecare nod va memora codul și titlul cărții, precum și numărul de exemplare existente în bibliotecă din cartea respectivă. Arborele va fi construit astfel încât să permită regăsirea unei cărți după codul său. Scrieți un program care, prin intermediul unui meniu, selectează în mod repetat, atâta timp cât utilizatorul dorește acest lucru, una din următoarele acțiuni:

aducerea unei cărți noi în bibliotecă;

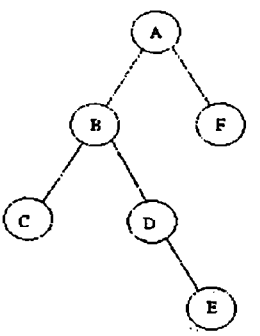
aducerea de noi exemplare dintr-o carte deja existentă;

listarea tuturor cărților din bibliotecă în ordinea crescătoare a codurilor.

92. Elaborați un algoritm iterativ care să traverseze un arbore binar în preordine, în inordine, în postordine.

Indicație. Pentru scrierea algoritmului se va folosi o stivă sub forma unui vector în care se vor memora nodurile la care va trebui să revenim.

93. Se dă o secvență ce conține litere și caracterele `(` `,` `)` și `$` astfel încât:



literele reprezintă informația din nod;

`(` definește un nivel inferior;

`,` separă arborescența stângă de cea dreaptă;

`)` s-a sfârșit nivel inferior;

`$` lipsește ramura corespunzătoare.

Să se construiască arborele binar atașat secvenței considerate.

Exemplu. Pentru secvența `(A(B(C,D($,E)),F)` se obține arborele din figura alăturată.

94. Fiind dat un arbore binar să se construiască secvența conform semnificațiilor de la problema anterioară.

95. Pentru evidența studenților unei facultăți se definește un arbore binar de căutare, în care fiecare nod va memora numărul matricol, numele și media anuală ale unui student. Căutarea în arbore se va face după numărul matricol al studenților. Scrieți un program care, prin intermediul unui meniu, selectează în mod repetat, atâta timp cât utilizatorul dorește acest lucru, una din următoarele acțiuni:

adăugarea de studenți;

afișarea studenților cu media cuprinsă între două valori `m1` și `m2` date;

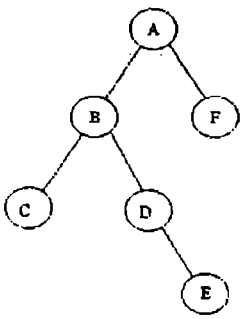
modificarea mediei unui student pentru care se cunoaște numărul matricol.

96. Mulţi algoritmi ce lucrează cu arbori preferă să viziteze de două ori fiecare nod în loc de o singură dată, folosind o combinaţie între preordine şi inordine denumită ordine duală. Traversarea unui arbore binar în ordinea duală este definită astfel: dacă arborele binar este vid nu se face nimic, în caz contrar:

   a) se vizitează rădăcina: pentru prima oară;
   b) se traversează sub-arborele stâng; în ordine duală;
   c) se vizitează rădăcina, pentru a doua oară;
   d) se traversează sub-arborele din dreapta, în ordine duală.

   Să se scrie un algoritm iterativ şi unul recursiv care să traverseze un arbore binar în ordine dală.

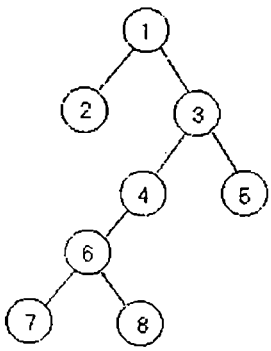
   Exemplu. Traversarea arborelui din figura de mai jos în ordine duală conduce la succesiunea: `A,B,C,B,D,E,A,F`.



97. Elaboraţi un algoritm iterativ care să traverseze un arbore binar în inordine fără a utiliza nici o stivă auxiliară. Se permite alterarea în orice nod a câmpurilor de legătură. Se poate utiliza dacă este nevoie un câmp `TAG` (de un singur bit) în fiecare nod, pentru memorare temporară.

98. Să se scrie un program care creează o listă liniară simplu înlănţuită cu cheile arborelui preluate pe nivele.

   Exemplu. Pentru arborele de mai jos, nodurile listei vor conţine, în ordine, următoarele chei: `1,2,3,4,5,6,7,8`.



99. Se poate observa că într-un arbore binar există mai multe referinţe de legătură nule decât alte pointere. Aceasta înseamnă că se iroseşte spaţiu de memorare şi există mai multe căi de a asigura o utilizare mai eficientă a memoriei.

O idee ar fi următoarea: se folosesc două "mărci" (TAG-uri) indicatoare pentru fiecare nod, care să indice cu ajutorul a doi biţi, dacă **st** sau **dr** sau ambii sunt nuli, iar spaţiul din memorie corespunzător nodurilor terminale se folosesc în alte scopuri.

O utilizare ingenioasă a acestui spaţiu este elaborarea aşa numiţilor arbori binari înseilaţi. De exemplu arborele înseilat al arborelui din figura a) este arborele din figura b).

Arborii binari înseilaţi se pot defini astfel:

Arborele neînseilat	**Arborele înseilat**	
p^.st=NIL	p^.tags="-"	p^.st=$p
p^.st=Q≠NIL	p^.tags="+"	p^.st=Q
p^.dr=NIL	p^.tagdr="-"	p^.dr=p$
p^.dr=Q≠NIL	p^.tagdr="+"	p^.dr=Q

unde am notat cu p$ succesorul lui p în inordine şi cu $p predecesorul lui p în inordine.

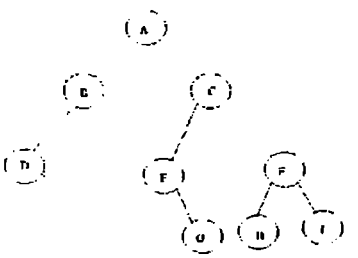


Fig. a)

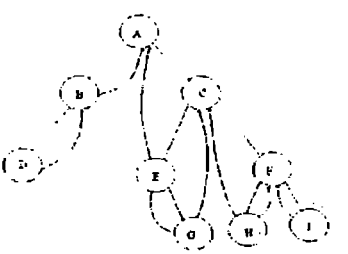


Fig. b)

a) elaboraţi un algoritm care determină succesorul în preordine al nodului p într-un arbore binar înseilat;

b) elaboraţi un algoritm care determină predecesorul în preordine al nodului p într-un arbore binar înseilat;

c) elaboraţi un algoritm care determină succesorul în postordine al nodului p într-un arbore binar înseilat;

d) elaboraţi un algoritm care determină predecesorul în postordine al nodului p într-un arbore binar înseilat;

e) elaboraţi u algoritm care înseilează un arbore binar neînseilat;

Notă. Se vor folosi algoritmii de la paşii anteriori;

f) elaboraţi un algoritm care adaugă un nod într-un arbore înseilat;

100. Există şi alt mod de înseilare a arborilor. Considerăm că reprezentarea va folosi în fiecare nod pe lângă informaţiile utile următoarele trei câmpuri: LTAG, LLINK, RLINK.

LTAG(p) are aceeaşi utilizare ca şi la arborii înseilaţi

LLINK(p) succesorul în preordine a lui p (fiul stâng)

RLINK(p) predecesorul în preordine a lui p (fiul drept).

a) elaboraţi un algoritm care determină succesorul în preordine al nodului p într-un astfel de arbore binar;

b) elaboraţi un algoritm care determină predecesorul în preordine al nodului p într-un astfel de arbore binar;

c) elaboraţi un algoritm care determină succesorul în postordine al nodului p într-un astfel de arbore binar;

d) elaboraţi un algoritm care determină predecesorul în postordine al nodului p într-un astfel de arbore binar;

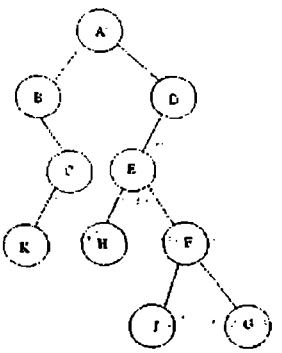
e) elaboraţi un algoritm care transformă un arbore binar neînseilat într-un astfel de arbore;

Notă. Se vor folosi algoritmii de la paşii anteriori;

f) elaboraţi un algoritm care adaugă un nod într-un astfel de arbore;

101. Elaboraţi un program care, fiind date tabelele INFO1 şi RLINK corespunzând reprezentării secvenţiale în preordine să formeze tabelele INFO2 şi GRAD corespunzând reprezentării secvenţiale în inordine.

Exemplu. Pentru arborele din figura de mai jos, algoritmul va transforma vectorii INFO1=(A,B,C,K,D,E,H,F,J,G). RLINK=(5,3,0,0,0,8,0,10,0,0) în INFO2=(B,K,C,A,H,E,J,F,G,D) şi GRAD=(1,0,1,2,0,2,0,2,0,1)



8.2.2. Arbori oarecare

102. Să se construiască o structură ramificată reprezentând un arbore genealogie pe linia unui singur părinte. Fiecare nod al arborelui conține numele persoanei, vârsta și trei câmpuri de legătură cu alte noduri:

`pcopil` – legătura spre nodul care reprezintă cel mai vârstnic cop il al persoanei respective;

`fmic` – legătura de fratele mai mic;

`fmare` - legătura spre fratele mai mare.

Să se regăsească o persoană dată în acest arbore genealogic și să se determine:

a) toți frații mai mari

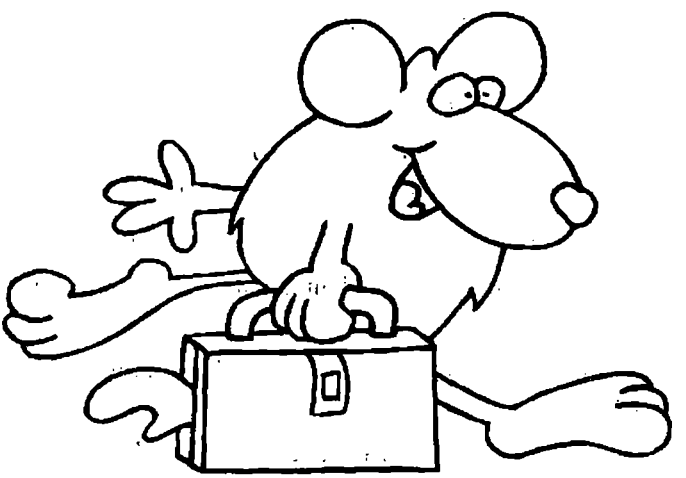
b) toți frații mai mici

c) bunicul

d) verișorii

e) nepoții de copii

f) nepoții de frate.



Bibliografie

1. A. Atanasiu – *Concursuri de informatică. Probleme propuse*, Editura Petrion, Bucureşti, 1995;
2. C. Cadar, M. Stroe – *Culegere de probleme şi programe PASCAL*, Editura Petrion, Bucureşti;
3. Th.H. Cormen, C.E. Leiserson, R.R. Rivest – *Introducere în algoritmi*, Computer Libris Agora, Cluj-Napoca, 2001;
4. D. Hrinciuc Logofătu – C++. *Probleme rezolvate şi algoritmi*, Editura Polirom, Iaşi, 2001;
5. C. Ivaşc, M. Prună, E. Mateescu – *Bazele informaticii*. Caiet de laborator, Editura Petrion, Bucureşti;
6. D. Knuth – *Tratat de programarea calculatoarelor*, Editura Tehnică, Bucureşti, 1974;
7. L. Livovschi, H. Georgescu – *Sinteza şi analiza algoritmilor*, Editura ştiinţifică şi enciclopedică, Bucureşti, 1986;
8. G. Mateescu, P.Fl. Moraru – *Informatica pentru liceu şi Bacalaureat – Materia de clasa a IX-a*, Editura Donaris, Sibiu, 2001;
9. G. Mateescu, P.Fl. Moraru – *Informatica pentru liceu şi Bacalaureat – Materia de clasa a X-a*, Editura Donaris, Sibiu, 2001;
10. G. Mateescu, P.Fl. Moraru, O. Sofron – *Informatica – Manual pentru clasa a X-a, varianta Pascal*, Editura Niculescu ABC, Bucureşti, 2000;
11. E. Mateescu, I. Maxim – *Arbori*, Editura Ţara Fagilor, Suceava, 1996;
12. MEC – Subiectele propuse la Bacalaureat şi variantele alternative, în anii 1999-2002;
13. V. Mitrana – *Provocarea algoritmilor. Probleme pentru concursurile de informatică*, Editura Agni, 1994;
14. M. Mocanu, Gh. Marian, C. Bădică – *333 probleme de programare*, Editura Teora, Bucureşti, 1994;
15. P.Fl. Moraru – *Bacalaureat. informatică*, Editura Petrion, Bucureşti, 2000;
16. M. Olteanu – *Probleme rezolvate în Pascal*, Editura Albastră, Cluj-Napoca, 1995;

17. O. Pătrăşcoiu, Gh. Marian, N. Mitroi – *Elemente de grafuri şi combinatorică. Metode, algoritmi şi programe*, Editura ALL, Bucureşti, 1994.

18. B. Pătruţ – *Aplicaţii C şi C++*, Editura Teora, Bucureşti, 1998;

19. D. Popescu Anastasiu – *Culegere de probleme de informatică pentru gimnaziu şi liceu*, Editura ALL Educaţional, Bucureşti, 2000;

20. D. Rancea – *Limbajul Pascal*, Manual pentru clasa a IX-a, Computer Libris Agora, Cluj-Napoca, 1997;

21. D. Rancea – *Limbajul Pascal. Algoritmi fundamentali*, Manual pentru clasa a X-a, Computer Libris Agora, Cluj-Napoca, 1999;

22. T. Sorin – *Turbo Pascal. Algoritmi şi limbaje de programare*, Manual pentru clasa a IX-a, Editura L&S Infomat, Bucureşti, 1996;

23. T. Sorin – *Tehnici de programare*, Manual pentru clasa a X-a, Editura L&S Infomat, Bucureşti, 1996;

24. T. Sorin – *Tehnici de programare 2*, Laborator clasa a XI-a, Editura L&S Infomat, Bucureşti;

25. D. Stoilescu – *Culegere de C/C++*, Editura Radial, Galaţi, 1998;

26. I. Tomescu – *Probleme de combinatorică şi teoria grafurilor*, Editura Didactică şi Pedagogică, Bucureşti, 1981.

27. N. Wirth – *Algorithms+Data Structures=Programs*, Prentice Hall, Inc 1976

28. *** – *Restructurarea perfecţionării profesorilor de informatică*, Ed. Computer Press Agora, 1998

29. *** – seria Gazeta de Informatică.

30. *** – seria GInfo